32/39
Т33

ТЕОРИЯ КОДИРОВАНИЯ

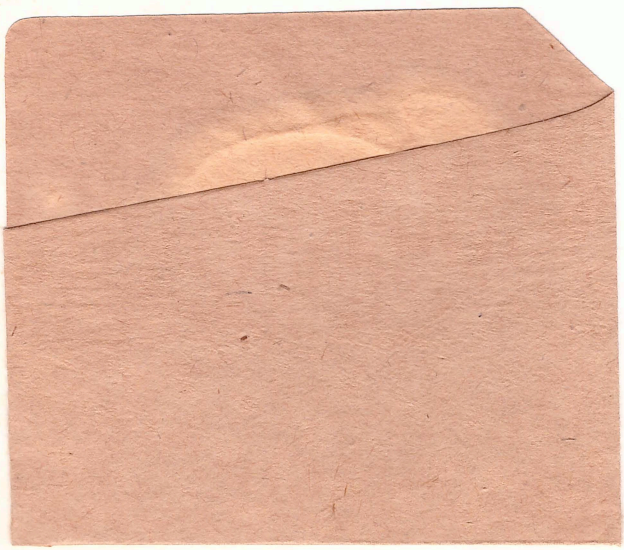
Т. КАСАМИ,
Н. ТОКУРА,
Ё. ИВАДАРИ,
Я. ИНАГАКИ

621.39
Т33

ТЕОРИЯ КОДИРОВАНИЯ

Т. КАСАМИ,
Н. ТОКУРА,
Ё. ИВАДАРИ,
Я. ИНАГАКИ

ТЕОРИЯ КОДИРОВАНИЯ
Т. КАСАМИ, Н. ТОКУРА, Ё. ИВАДАРИ, Я. ИНАГАКИ



ИЗДАТЕЛЬСТВО
МИР

符号理論

大阪大学教授 工学博士 嵩 忠雄
大阪大学助教授 工学博士 都倉信樹
日本電気株式会社 工学博士 岩垂好裕
名古屋大学助教授 工学博士 稲垣康善
共著

情報工学講座 4

コロナ社

Т. КАСАМИ,
Н. ТОКУРА,
Ё. ИВАДАРИ,
Я. ИНАГАКИ

ТЕОРИЯ КОДИРОВАНИЯ

Перевод с японского
А. В. КУЗНЕЦОВА

под редакцией
Б. С. ЦЫБАКОВА И С. И. ГЕЛЬФАНДА

ИЗДАТЕЛЬСТВО «МИР»
МОСКВА 1978

УДК 681.32

В книге систематически излагается теория кодов, исправляющих ошибки, и рассматривается их применение в системах связи и вычислительной технике. В последние годы интерес к вопросам использования кодовых методов защиты от ошибок значительно возрос в связи с развитием сетей передачи данных и особенно сетей с коммутацией пакетов. В книге рассматриваются важнейшие классы кодов: блоковые, сверточные и арифметические. Приводятся последние достижения теории кодирования. Подробно обсуждаются возможности использования кодов в практических системах.

Книга полезна специалистам, работающим в области систем связи, вычислительной техники и автоматизированных систем управления, математикам и кибернетикам, интересующимся теорией кодирования, а также аспирантам и студентам соответствующих специальностей.

Редакция литературы по новой технике

К $\frac{30401\text{-}374}{041(01)\text{-}78}$ 146-77

© Т. Касами, Н. Токура, Ё. Ивадари, Я. Инагаки, 1975
© Перевод на русский язык, «Мир», 1978

Предисловие редакторов русского издания

Предлагаемая советскому читателю книга посвящена одному из центральных разделов теории информации — кодам, исправляющим ошибки. Она написана японскими учеными Т. Касами, Н. Токура, Ё. Ивадари и Я. Инагаки. Первые два автора приобрели мировую известность благодаря своим работам по теории кодов.

Из переводных книг по теории кодирования наибольшую известность в нашей стране получила книга У. Питерсона «Коды, исправляющие ошибки» (издательство «Мир», 1964). В 1976 г. издательство «Мир» выпустило перевод второго издания этой книги, подготовленного У. Питерсоном совместно с Э. Уэлдоном. Во многих отношениях эта замечательная книга основана на работах, выполненных до 1970 г. Однако в семидесятых годах учеными разных стран был получен ряд новых результатов, существенно обогативших теорию кодирования и в известном смысле изменивших некоторые ее общие концепции. Данная книга японских авторов включает наиболее важные из этих результатов, такие, как коды Юстесена, коды Гоппы, каскадные коды, циклические арифметические коды и др.

Теория кодирования является прикладной наукой. Она черпает задачи из техники связи, радиолокации, измерительной, вычислительной и управляющей техники. Поэтому важное значение имеют применимость получаемых результатов, их конкурентоспособность по сравнению с некодовыми методами защиты от ошибок. Практические достижения теории кодирования сейчас хорошо известны. Однако наиболее широкое применение получили лишь самые примитивные способы кодирования, такие, как проверка на четность, коды Хэмминга, циклические проверки для обнаружения ошибок, простейшие сверточные коды и т. п. С точки зрения специалистов по теории кодирования, отставание в использовании более мощных и нетривиальных кодовых конструкций происходит из-за сложности восприятия инженерами математического аппарата теории и из-за недостаточной разработанности вопросов взаимосвязи кодирования и модуляции.

Книга японских авторов помогает преодолеть это отставание. Авторы излагают математические вопросы весьма обстоятельно, неторопливо, сопровождая текст многочисленными примерами и задачами учебного характера. Восточная

медлительность здесь выступает удачной альтернативой западной лаконичности, образцом которой служат произведения Н. Бурбаки. В книгу помещена специальная глава (гл. 7), посвященная реализации и применениям кодов. В этой главе рассматриваются принципы построения кодирующих и декодирующих устройств для циклических и сверточных кодов. Приводятся способы использования двоичных и других дискретных кодов в реальных каналах, которые по своей физической природе являются аналоговыми. Обсуждается проблема согласованного выбора методов кодирования и модуляции. Вводится новый параметр, измеряющий выигрыш при использовании кодирования в канале. Эта не претендующая на математическую строгость глава призвана сыграть важную роль связующего звена между теорией кодов и ее практическими применениями.

Хочется отметить, что теория информации сейчас находится в процессе развития. Поэтому ее достижения и тенденции за время, прошедшее после выхода в свет данной книги, не могли быть в ней отражены. К вопросам, не включенным в книгу, относятся, в частности, коды для многоисточниковых, широковещательных и многосторонних каналов, коды, исправляющие дефекты и ошибки, и коды, повышающие скрытность передачи. В настоящее время с этими достижениями можно ознакомиться лишь по журнальным и обзорным статьям.

При переводе и редактировании были исправлены некоторые опечатки, имеющие место в японском издании. Часть из них была любезно указана нам авторами, которым приносим глубокую благодарность.

С. И. Гельфанд, Б. С. Цыбаков

Предисловие авторов

К настоящему времени по теории кодирования вышли следующие монографии: книга Питерсона [1] и ее расширенный вариант — монография Питерсона и Уэлдона [2], оригинальная книга по алгебраической теории кодирования Берлекэмпа [3], книга Ван-Линта, систематизирующая алгебраическую теорию кодирования [4], прекрасное введение в теорию кодирования Лина [5] и большой труд Миякавы, Ивадари и Имаи [6]. Однако уже после выхода в свет этих книг Гоппа, Юстесен и др. получили ряд новых важных результатов.

В теории кодирования можно выделить следующие три раздела: теорию алгебраических кодов, теорию сверточных кодов и теорию кодов, используемых в арифметических устройствах. Все эти разделы тесно связаны друг с другом, но в то же время характеризуются собственным подходом к проблеме кодирования. В данной книге этим разделам посвящены соответственно главы 1—4; 5, 6; 8, 9. Каждый раздел, как правило, начинается введением основных понятий и заканчивается изложением сравнительно новых результатов. Авторы не претендуют на исчерпывающее изложение всего материала теории кодирования, скорее, они преследуют цель рассмотреть все наиболее важные моменты в развитии методов теории.

В гл. 7 подробно исследуются различные вопросы, связанные с применением теории кодирования. При первом знакомстве с теорией кодирования затруднения, как правило, возникают там, где используются понятия и результаты теории конечных полей. Поэтому в эту книгу было решено включить главу, в которой кратко рассматриваются основные понятия и результаты теории конечных полей, используемые в теории кодирования. Для понимания содержания книги достаточно знакомства с университетским курсом математики.

Разд. 1.1—1.4 гл. 1, а также гл. 3, 4 написаны Касами, гл. 2 — Токура, разд. 1.5 гл. 1 и гл. 5—7 — Ивадари, гл. 8 и 9 — Инагаки. Главы 1—4, 5—7 и 8, 9 можно читать почти независимо. Кроме того, читатели, знакомые с конечными полями, могут пропустить гл. 2 и приступить к чтению последующих глав, обращаясь к гл. 2 лишь по мере необходимости.

В гл. 1 вводятся основные понятия теории кодирования, в частности понятия, связанные с блоковыми кодами, определяется ряд границ, характеризующих предельные характеристики

блоковых кодов; в разд. 1.5 методом Галлагера доказана теорема кодирования Шеннона, которая в свое время послужила отправной точкой теории кодирования.

В гл. 2 рассматриваются группы, кольца, поля, векторные пространства, многочлены и другие понятия и результаты теории конечных полей в том объеме, который необходим для чтения статей по теории кодирования. Многие результаты теории кодирования формулируются в виде задач или утверждений.

В гл. 3 рассматриваются свойства линейных кодов и методы их кодирования и декодирования, свойства циклических кодов, являющихся подклассом линейных кодов, а также методы задания таких кодов. При этом подчеркивается значение алгебраической теории кодирования, которая изучает классы кодов с определенной алгебраической структурой и использует эту структуру для детального описания свойств кодов и построения для них эффективных алгоритмов кодирования и декодирования.

Алгебраическая теория кодирования содержит много результатов, которые заслуживают подробного рассмотрения, и поэтому выбрать материал для данной книги было довольно трудно. В гл. 4 основной акцент был сделан на пояснение различных методов теории кодирования, в соответствии с чем и отбирался материал для этой главы. В эту главу включены наиболее детально исследованные к настоящему времени БЧХ-коды и методы их декодирования, важные с практической точки зрения методы мажоритарного декодирования, играющие важную роль в теории многочлены Матсона — Соломона, полиномиальные коды, содержащие в качестве своих подкодов БЧХ-коды, некоторые классы кодов, допускающих мажоритарное декодирование, их двойственные коды, каскадные коды, строящиеся на основе нескольких кодов с меньшими значениями параметров, каскадные коды Юстесена, представляющие собой первый класс кодов, свойства которых не ухудшаются в области больших значений параметров. Она содержит также интенсивно исследуемые в настоящее время коды Гоппы (вообще говоря, не являющиеся циклическими), включающие как подкласс БЧХ-коды, методы декодирования кодов Гоппы [1]) и несколько отличающиеся от кодов, перечисленных выше, коды, исправляющие пачки ошибок. К сожалению, в этой книге не удалось рассмотреть квадратично-вычетные коды [2, 3], нелинейные коды [6], обстоятельные работы, посвященные исследованию струк-

[1]) Уже после того, как книга была написана, Хелгерт, Сугияма — Касахара, Чень — Чоу опубликовали метод задания кодов Гоппы, несколько отличный от метода Гоппы, а также итеративный алгоритм декодирования последних.

туры весов [3, 4], проблему синхронизации [2] и некоторые другие важные вопросы.

Читателям, которые впервые хотели бы ознакомиться с основами теории блоковых кодов, рекомендуется изучить следующие разделы: 1.1—1.3, 1.4.1, помеченные знаком * разделы гл. 2, 3.1—3.3, 3.5—3.7, 4.1, 4.2.1, 4.3, 4.8.

Гл. 5 и 6 посвящены сверточным кодам. Рассматриваемые в начале гл. 5 коды, которые допускают пороговое декодирование, имеют простые алгоритмы декодирования и могут быть легко реализованы, что очень важно с практической точки зрения. Методы построения и декодирования таких кодов главным образом являются алгебраическими и основаны на теории автоматов. Особое внимание в этой главе уделяется возможности практической реализации декодирования. Простое декодирование допускают рассматриваемые здесь коды, исправляющие независимые ошибки, коды, исправляющие пачки ошибок, диффузные коды, исправляющие независимые ошибки и пачки ошибок, равномерные коды, похожие на блоковые коды, образованные последовательностями максимальной длины.

При последовательном декодировании, которое рассматривается в гл. 6, число операций изменяется в зависимости от уровня шума, действующего в канале. Благодаря этому последовательное декодирование по своим характеристикам приближается к декодированию по максимуму правдоподобия и широко используется в реальных системах космической связи. Алгебраические методы для анализа этого метода декодирования не подходят. С помощью вероятностных методов для последовательного декодирования выводятся различные границы, в частности верхние и нижние границы для распределения выполняемых декодером операций, вероятности переполнения буфера, вероятности ошибки при декодировании и др. Анализ последовательного декодирования оказывается весьма сложным. Кроме алгоритмов последовательного декодирования, в этой главе рассматривается также алгоритм декодирования Витерби, который представляет собой алгоритм декодирования по максимуму правдоподобия. Сравнение сверточных кодов с блоковыми кодами показывает, что первые при декодировании с помощью алгоритма Витерби имеют лучшие характеристики; это один из важных выводов теории кодирования. Кроме того, в этой главе рассматривается гибридный метод декодирования, представляющий собой объединение последовательного и алгебраического декодирования, а также исследуется структура расстояний сверточных кодов.

В гл. 7 сначала описываются основные компоненты, входящие в состав кодирующих и декодирующих устройств, а также схемы кодирующих и декодирующих устройств для некоторых

кодов. Вторая часть этой главы посвящена проблемам, связанным с применением теории кодирования. Для развития теории кодирования использовались в основном методы современной алгебры и различные методы получения оценок; из-за сложности этих методов существуют трудности в понимании связи теории с теми реальными задачами, которые она призвана решать. Чтобы преодолеть эти трудности, в гл. 7 исследуется адекватность математических моделей их реальным прообразам и исследуются границы применимости теории кодирования. Основными объектами применения теории кодирования являются системы связи и вычислительные машины. В этой главе введено понятие выигрыша при применении кодирования в системе связи, дано несколько примеров систем связи с кодированием и показана целесообразность введения этого понятия. В разделах, посвященных применениям теории кодирования в вычислительных машинах, рассмотрены коды, исправляющие ошибки оператора при вводе информации в машину, а также коды, допускающие быстрое декодирование.

Тем, кто хочет ознакомиться лишь с основами сверточных кодов, рекомендуем прочитать разделы 5.1—5.8, 6.1 и 6.2. Читатели, которые интересуются лишь применениями теории кодирования, могут читать гл. 7 независимо.

В гл. 8 и 9 рассматриваются коды, предназначенные для обнаружения и исправления арифметических ошибок, возникающих в арифметических устройствах, в частности AN-коды. Кодовыми словами AN-кодов являются двоичные представления целых чисел, кратных числу A. Эти коды отличаются от всех кодов, рассматривавшихся до гл. 8. Поскольку для изложения теории таких кодов требуются дополнительные сведения о свойствах целых чисел и их двоичных представлений, то в начале гл. 8 приводятся основные понятия и результаты теории целых чисел, используемые в дальнейшем. Далее в гл. 8 дается определение AN-кода, описываются особенности ошибок, возникающих при выполнении арифметических операций, в частности при сложении и вычитании чисел, вводится арифметическое расстояние, играющее в теории арифметических кодов ту же роль, что и расстояние Хэмминга в теории алгебраических кодов, описывается несколько классов кодов, исправляющих одиночные и кратные ошибки, и исследуются их свойства.

Гл. 9 посвящена циклическим AN-кодам. Как и алгебраические циклические коды, циклические AN-коды обладают рядом достоинств и особых свойств. С помощью разложения целых чисел на простые множители и приведенной системы вычетов по ее подгруппе циклические AN-коды можно разложить на строго циклические коды и таким образом исследовать их структуру. Кроме того, это разложение указывает на систематический ме-

тод построения циклических AN-кодов и позволяет оценить распределение весов кодовых слов. В конце главы кратко рассмотрены AN-коды, исправляющие пачки ошибок, арифметические коды с несколькими модулями, q-ичные AN-коды и проведено сравнение AN-кодов и алгебраических кодов.

В заключение нам хочется поблагодарить С. Адзуми за составление табл. А. 1 и А. 2 приложения и внесение исправлений в гл. 1—4.

Авторы

Литература

1. Peterson W. W., Error-Correcting Codes, MIT Press, Cambridge, Mass. 1961.
2. Peterson W. W., Weldon E. J., Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972; есть русский перевод: Питерсон У., Уэлдон Э., Коды, исправляющие ошибки, изд-во «Мир», 1976.
3. Berlekamp E. R., Algebraic Coding Theory, McGraw-Hill, N. Y., 1968; есть русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, изд-во «Мир», 1971.
4. Van Lint J. H., Coding Theory, Springer-Verlag, Berlin, Heidelberg, N. Y., 1971.
5. Lin S., Introduction to Error Correcting Codes, Prentice-Hall, Englewood Cliffs, N. J., 1970.
6. Миякава, Ивадари, Иман, Теория кодирования, изд-во „Сёкодо", 1973.

1. Основные понятия теории кодирования

1.1. Коды, обнаруживающие и исправляющие ошибки

Если взять достаточно длинное предложение английского или японского текста и исказить его, заменив в некоторых местах одни буквы на другие, исключив ряд букв (символов) или добавив новые, то, используя знание структуры отдельных слов и предложения в целом, предыдущий и последующий тексты, а также наши обычные познания о предмете, о котором идет речь, можно в большом числе случаев полностью восстановить исходное предложение или по крайней мере безошибочно уловить его смысл. Это показывает, что английский, японский и другие естественные языки обладают очень большой избыточностью. В технике также часто устанавливают по два или три одинаковых устройства, чтобы при выходе из строя одного из них его можно было заменить другим. В некоторых случаях с помощью одного устройства выполняются дважды одни и те же вычисления, и если результаты этих вычислений совпадают, то принимается решение о том, что вычисления выполнены без ошибок. Если же результаты вычислений не совпадают, то это означает, что в процессе вычислений произошла ошибка. Вычисления проводятся еще раз, и обнаруженная ошибка устраняется. Во всех рассмотренных выше случаях наличие избыточности позволяет провести обнаружение ошибок и повысить надежность устройств.

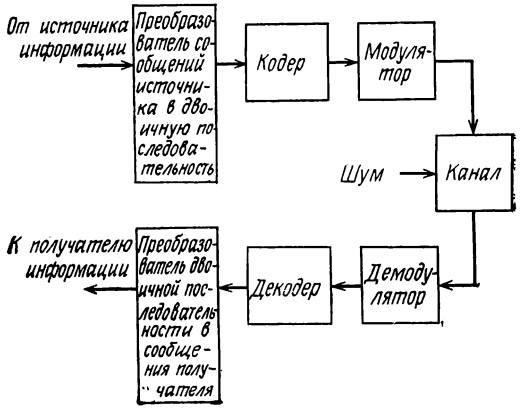
Однако проблема состоит не в том, чтобы просто повысить надежность за счет введения очень большой избыточности, а в том, как с помощью по возможности меньшей специальным образом вводимой избыточности достичь нужной степени надежности.

В некоторых предложениях, взятых из естественного языка, достаточно сильно исказить небольшое число букв (например, считать эти буквы неопределенными), чтобы смысл был полностью искажен. Известно [1][1]), что избыточность английского языка достигает 70%, однако нельзя сказать, что избыточность вводится в английский текст оптимально с точки зрения возможности исправления и обнаружения ошибок. Основной задачей теории кодирования в настоящее время является повышение надежности систем связи и вычислительных систем

---

[1]) Цифры в квадратных скобках указывают номер работы в списке литературы, помещенном в конце книги.

с помощью целенаправленного эффективного введения избыточности в процессе представления информации (в процессе преобразования информации). Введение избыточности приводит к снижению количества сообщений, которые могут быть переданы или обработаны за определенный период времени, а кроме того, предполагает использование в системе дополнительных устройств для целенаправленного введения избыточности (кодеров), устройств для обнаружения и исправления возникающих ошибок (декодеров) и ряда других дополнительных устройств. Само собой разумеется, что проектирование таких избыточных систем должно выполняться с учетом стоимости этих дополнительно вводимых устройств (см. гл. 7). Поскольку



Фиг. 1.1. Модель системы связи.

в теории обычно рассматриваются идеализированные модели реальных систем, в которых из большого числа факторов, влияющих на поведение реальной системы, учитывается лишь часть, то теоретические результаты должны использоваться для решения задач, связанных с построением реальных систем, только после того, как будут определены границы применимости этих теоретических результатов.

На фиг. 1.1 показана типичная модель системы связи, в которой используются коды, исправляющие ошибки [2]. В этой модели каналом является простейший двоичный канал. Эта же модель может быть использована и в качестве модели запоминающего устройства (например, запоминающего устройства на магнитной ленте или интегральной памяти), если среду, в которой хранится информация, рассматривать как канал связи. Поступающая от источника информация вначале с помощью преобразователя «источник информации — двоичная последовательность» преобразуется в последовательность двоичных сим-

волов, которая подается на вход кодера, где в нее вводится избыточность. Символы с выхода кодера с помощью модулятора преобразуются в сигналы, которые могут быть переданы по каналу. Эти сигналы поступают в канал, где они обычно искажаются шумами. На приемном конце искаженные сигналы канала с помощью демодулятора преобразуются в последовательность двоичных символов, содержащую избыточные символы. Используя эту избыточность, декодер обнаруживает и исправляет возникшие ошибки. Двоичная последовательность символов на выходе декодера уже не является избыточной. Если воздействие шума в канале не слишком сильное и декодер может исправить все возникшие ошибки, то двоичная последовательность на выходе декодера будет совпадать с двоичной последовательностью на выходе преобразователя «источник информации — двоичная последовательность». И наконец, двоичная последовательность, получающаяся на выходе декодера, преобразуется в сигналы, приемлемые для получателя информации, и передается последнему.

Ниже рассматривается несколько примеров простых кодов, обнаруживающих или исправляющих ошибки.

Пример 1.1. Рассмотрим канал, по которому за время t_0 может быть передан один импульс типа «0» или типа «1». Допустим, что при передаче могут происходить случайные ошибки, так что импульсы 0 и 1 могут быть приняты с некоторой вероятностью $1 - p_0$ соответственно как 1 и 0. Предположим, что ошибки при передаче различных импульсов возникают независимо. Пусть информацией, подлежащей передаче, является последовательность двоичных символов. Каждый символ a_i этой последовательности передается по каналу с помощью последовательности из пяти импульсов 00000, если $a_i = 0$, и 11111, если $a_i = 1$. На приемном конце принимаемая последовательность импульсов разбивается на группы по пять импульсов, называемые блоками, в соответствии с разбиением передаваемой последовательности. Если в принятом блоке содержится два и менее импульса 0, то принимается решение о том, что передавался символ $a_i = 1$. Если в принятом блоке содержится три и более импульсов 0, то считается, что $a_i = 0$. Очевидно, что переданный символ a_i будет декодирован верно тогда и только тогда, когда число ошибочно принятых импульсов в соответствующем блоке не больше двух. Следовательно, вероятность ошибочного декодирования любого символа a_i равна[1])

[1]) Через $\binom{n}{m}$ обозначено число сочетаний из n по m. — *Прим. ред.*

$\sum_{i=3}^{5} \binom{5}{i} p_0^{5-i}(1-p_0)^i$. Этот метод передачи является простым, но для передачи одного бита информации он требует интервала времени длительностью $5t_0$. В то же время существуют коды (см. теорему кодирования из разд. 1.5), которые позволяют передавать по тому же самому каналу один бит информации в среднем за меньший интервал времени с вероятностью ошибки, не превосходящей указанную выше величину. При этом, однако, процедуры кодирования и декодирования оказываются более сложными. В тех случаях, когда «убытки» из-за ошибочного декодирования оказываются больше убытков, возникающих из-за того, что часть символов a_i вообще не декодируется и считается неизвестной (как это имеет место, например, в системах с переспросом), часто оказывается более целесообразным использовать другой алгоритм декодирования. Этот алгоритм состоит в том, что если в блоке нет символов 1 или содержится только один символ 1, то считается, что $a_i = 0$, если же в блоке содержатся четыре или пять символов 1, то считается, что $a_i = 1$. Во всех остальных случаях решение не принимается. Для этого алгоритма вероятность ошибочного декодирования равна $\sum_{i=4}^{5} \binom{5}{i} p_0^{5-i}(1-p_0)^i$, а вероятность отказа от декодирования равна $\sum_{i=2}^{3} \binom{5}{i} p_0^{5-i}(1-p_0)^i$.

Пример 1.2. В вычислительных машинах информация обычно представляется в виде двоичных последовательностей. При этом используется несколько способов представления десятичных цифр 0, 1 ..., 9. Одним из таких способов является код «два из пяти», показанный в табл. 1.1. Любая из цифр в этом коде представляется в виде двоичной последовательности длины 5, в которой два из пяти символов всегда равны 1 (заметим, что $\binom{5}{2} = 10$). При возникновении одиночной ошибки, т. е. при искажении одного из двоичных символов, последовательность содержит один или три символа, равных 1, а это позволяет обнаружить возникновение ошибки. В общем случае для представления каких-либо $\binom{n}{w}$ сообщений могут использоваться двоичные последовательности длины n, каждая из которых содержит w символов 1. Совокупность таких двоичных последовательностей называется *равновесным кодом*. Коды этого типа оказываются эффективными, в частности, в тех случаях, когда вероятность возникновения ошибки типа «переход символа 0 в

Таблица 1.1

Пример представления десятичных цифр

Десятичные цифры	Код «2 из 5»	8 4 2 1	Символ проверки на четность
0	0 0 0 1 1	0 0 0 0	0
1	0 0 1 0 1	0 0 0 1	1
2	0 0 1 1 0	0 0 1 0	1
3	0 1 0 0 1	0 0 1 1	0
4	0 1 0 1 0	0 1 0 0	1
5	0 1 1 0 0	0 1 0 1	0
6	1 0 0 0 1	0 1 1 0	0
7	1 0 0 1 0	0 1 1 1	1
8	1 0 1 0 0	1 0 0 0	1
9	1 1 0 0 0	1 0 0 1	0

символ 1» значительно больше (или значительно меньше), чем вероятность возникновения ошибки типа «переход символа 1 в символ 0» (т. е. при так называемых несимметричных ошибках), а также в тех случаях, когда «убытки» из-за ошибочного принятия символа 1 вместо символа 0 значительно больше (меньше) убытков при принятии символа 0 вместо символа 1 [17—19]. Рассмотренные здесь коды являются примерами так называемых нелинейных кодов.

Пример 1.3. Предположим, что некоторые 2^{n-1} или менее сообщений представлены в виде двоичных последовательностей $(x_1, x_2, \ldots, x_{n-1})$ длины $n-1$ и это представление взаимно однозначно. Пусть

$$x_n = x_1 + x_2 + \ldots + x_{n-1}, \tag{1.1}$$

где в данном случае знак $+$ означает сложение по модулю 2, т. е. $x_n = 0$, если сумма в правой части (1.1) четна, и $x_n = 1$, если она нечетна. Предположим, что вместо последовательностей $(x_1, \ldots, x_{n-1})$ длины $n-1$ используются последовательности $(x_1, \ldots, x_n)$ длины n. Допустим, что в процессе передачи или обработки одной из таких последовательностей, содержащей один избыточный двоичный символ, возникла ошибка в одном из двоичных символов. Как видно из формулы (1.1), в неискаженной последовательности $(x_1, \ldots, x_n)$ число символов 1 всегда четно, а при возникновении ошибки в одном из символов это число становится нечетным. Это позволяет обнаруживать возникновение одиночных ошибок. Символ x_n

называют *символом проверки на четность*. Рассмотренный здесь метод обнаружения одиночных ошибок благодаря своей простоте применяется очень широко. В табл. 1.1 показан способ представления десятичных цифр с помощью пяти двоичных символов, первые четыре из которых являются двоичным разложением соответствующей десятичной цифры, а пятый — символом проверки на четность.

Пример 1.4. Предположим, что имеется 16 сообщений, каждому из которых взаимно однозначно поставлена в соответствие двоичная последовательность (x_3, x_5, x_6, x_7) длины 4. Вместо последовательности (x_3, x_5, x_6, x_7) для представления сообщений будем использовать двоичную последовательность $(x_1, x_2, \ldots, x_7)$ длины 7, символы x_1, x_2 и x_4 которой определим следующим образом:

$$\begin{bmatrix} 1&0&1&0&1&0&1 \\ 0&1&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = 0, \quad \text{т. е.} \quad \begin{cases} x_1 = x_3 + x_5 + x_7, \\ x_2 = x_3 + x_6 + x_7, \\ x_4 = x_5 + x_6 + x_7. \end{cases} \tag{1.2}$$

Здесь знак $+$ означает сложение по модулю 2, а остальные обозначения имеют тот же смысл, что и в обычных матричных соотношениях. Поскольку в процессе передачи или хранения последовательности $(x_1, x_2, \ldots, x_7)$ могут возникнуть ошибки, то принимаемая последовательность $(x_1', x_2', \ldots, x_7')$, вообще говоря, будет отлична от исходной. Рассмотрим задачу восстановления исходной последовательности $(x_1, x_2, \ldots, x_7)$ по известной последовательности $(x_1', x_2', \ldots, x_7')$. Предположим, что вероятность возникновения двух и более ошибок в двоичных символах одной последовательности длины 7 настолько мала, что этим событием можно пренебречь. Положим

$$x_i + x_i' = e_i,$$

где знак $+$ обозначает сложение по модулю 2. Если $e_i = 1$, то это означает, что в i-м двоичном символе произошла ошибка; если $e_i = 0$, то i-й двоичный символ передан без ошибок. Используя матрицу из левой части (1.2), определим r_1, r_2 и r_3 сле-

дующим образом:

$$\begin{bmatrix} 1&0&1&0&1&0&1 \\ 0&1&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \\ x_5' \\ x_6' \\ x_7' \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}. \tag{1.3}$$

Из формул (1.2) и (1.3) получаем

$$\begin{bmatrix} 1&0&1&0&1&0&1 \\ 0&1&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}. \tag{1.4}$$

По предположению только один из символов $e_1, e_2, \ldots, e_7$ может быть равен 1. При отсутствии ошибок, когда $e_1 = e_2 = \ldots = e_7 = 0$, имеем $r_1 = r_2 = r_3 = 0$.

Далее рассмотрим случай, когда $e_i = 1$, а остальные шесть символов e_j, $j \neq i$, равны 0. Заметим, во-первых, что все столбцы матрицы

$$P = \begin{bmatrix} 1&0&1&0&1&0&1 \\ 0&1&1&0&0&1&1 \\ 0&0&0&1&1&1&1 \end{bmatrix}$$

из левой части (1.4) различны, а во-вторых, что k-й столбец этой матрицы представляет собой двоичную запись номера k (самый младший разряд двоичного представления числа k является верхним элементом столбца). При сделанных выше предположениях вектор-столбец $\begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$ совпадает с i-м столбцом матрицы P. Следовательно, номер i искаженного двоичного символа может быть найден следующим образом: $i = r_1 + 2r_2 + 4r_3$. С помощью равенства $x_i = x_i' + e_i$ находим правильные двоичные символы $x_1, \ldots, x_7$. Множество двоичных последовательностей $(x_1, x_2, \ldots, x_7)$, удовлетворяющих соотношению (1.2) и

называемых обычно *кодовыми векторами*, называется *кодом Хэмминга* длины 7 [3].

Утверждение 1.1. *Если последовательности* $X = (x_1, x_2, \ldots, x_7)$ *и* $Y = (y_1, y_2, \ldots, y_7)$ *удовлетворяют соотношению* (*1.2*) (*т. е.* $PX^T = 0$ и $PY^T = 0$) и $X \neq Y$, *то число* $d(X, Y)$ *значений индекса* i, *таких, что* $x_i \neq y_i$ (*это число называется расстоянием Хэмминга* [*3*] *между векторами* X *и* Y), *не меньше трех* (*символ* T *означает транспонирование*).

Краткое доказательство. Число ненулевых компонент вектора называют *весом* этого вектора. Пусть Z — произвольный вектор веса 1 или 2. Так как все столбцы матрицы P различны и ни один из них не является нулевым, то $PZ^T \neq 0$. Вместе с тем по условиям утверждения $P(X - Y)^T = 0$. Наконец, заметим, что вес вектора $X - Y$ совпадает с $d(X - Y, 0)$ и $d(X - Y, 0) = d(X - Y)$ (здесь 0 — нулевой вектор).

Утверждение 1.2. *Для любого двоичного вектора* Y *длины 7 найдется такой кодовый вектор* X, *что* $d(X, Y) \leqslant 1$.

Краткое доказательство. Если Y — кодовый вектор, то $X = Y$. Предположим, что Y не является кодовым вектором. Пусть $PY^T = r$ (r — двоичный вектор длины 3, $r \neq 0$). Среди столбцов матрицы P существует ровно один столбец, в точности совпадающий с r. Пусть i — номер этого столбца. Возьмем в качестве вектора X вектор, отличающийся от Y только i-й компонентой. Тогда $PX^T = 0$ и $d(X, Y) = 1$.

Для представления в двоичной записи K различных сообщений достаточно $[\log_2 K]$ двоичных символов ($[A]$ — минимальное целое число, не меньшее A). Однако, как показывают приведенные выше примеры, использование некоторого числа дополнительных избыточных символов позволяет осуществить исправление или обнаружение части возникающих ошибок или упростить схемы, осуществляющие обработку этих сообщений. Число $n - [\log_2 K]$, где n — общее число двоичных символов, используемых для представления каждого сообщения, называют *числом проверочных* или *избыточных символов*. Естественно, что всегда хотелось бы достичь нужных свойств кода (например, способности исправлять так называемые одиночные ошибки, т. е. ошибки в одном двоичном символе), используя по возможности меньшее число избыточных символов.

Утверждение 1.3. *В случае* $K = 16$ *никакой код, имеющий два или менее проверочных символа, не может исправлять одиночные ошибки. В этом смысле код Хэмминга длины 7 является наилучшим среди кодов, исправляющих одиночные ошибки.*

Краткое доказательство. Для того чтобы код исправлял одиночные ошибки, множества векторов, находящихся на расстоя-

нии Хэмминга 1 и менее от каждого кодового слова, не должны пересекаться. Если длина кода равна n, то число векторов, находящихся на расстоянии 1 и менее от кодового слова, равно $n+1$. Так как общее число двоичных векторов длины n равно 2^n, то для любого кода, исправляющего одиночные ошибки, должно выполняться неравенство $(n+1)K \leqslant 2^n$, откуда следует справедливость доказываемого утверждения.

По матрице P из примера 1.4 построим матрицу P' следующим образом:

$$P' = \begin{bmatrix} & & 0 \\ & P & 0 \\ & & 0 \\ \underbrace{1\ 1\ \ldots\ 1}_{8} \end{bmatrix}.$$

Совокупность всех двоичных векторов X длины 8, удовлетворяющих условию $P'X^T = 0$, называется *расширением кода* из примера 1.4.

Утверждение 1.4. *Расширение кода из примера 1.4 исправляет одиночные ошибки и в то же время обнаруживает двойные ошибки (т. е. ошибки в любых двух компонентах кодового вектора).*

Идея доказательства. Вначале, используя утверждение 1.1, нужно показать, что расстояние Хэмминга между любыми двумя различными кодовыми векторами не меньше четырех. Основываясь на этом, далее следует доказать, что рассматриваемый код исправляет одиночные и обнаруживает двойные ошибки (в общем случае связь между минимальным значением расстояния Хэмминга и способностью кода исправлять и обнаруживать ошибки устанавливается в разд. 1.3).

При изучении корректирующей способности кода Хэмминга из примера 1.3 и его расширения из утверждения 1.4 для нас существенным было лишь то, что все столбцы матрицы P различны и все они являются ненулевыми. Аналогично можно построить коды для любого $K = 2^m$. В общем случае коды Хэмминга и их расширения рассматриваются в задаче 3.2, примере 3.5 и разд. 4.1.

1.2. Блоковые коды. Систематические коды

Рассмотрим q-ичный канал ($q \geqslant 2$), по которому могут передаваться сигналы (импульсы) q типов определенной длительности. Для обозначения этих сигналов будем использовать q

различных символов, которые назовем символами канала; множество q символов канала обозначим через V_1. Пусть V_n — множество всех последовательностей длины n из символов V_1, т. е. $V_n = \{(a_1, a_2, \ldots, a_n) \mid a_i \in V_1, \ 1 \leqslant i \leqslant n\}$. Расстояние Хэмминга $d(A, B)$ между последовательностями $A = (a_1, a_2, \ldots, a_n)$ и $B = (b_1, b_2, \ldots, b_n)$ длины n определим как число пар компонент a_i и b_i, таких, что $a_i \neq b_i$. Заметим, что это расстояние зависит лишь от того, совпадают или нет соответствующие символы последовательностей A и B. Расстояние Хэмминга может быть использовано в системах модуляции с ортогональными сигналами [4]. Это связано с тем, что, когда q символов канала представляют q взаимно ортогональных сигналов с равной энергией, а шум в канале является аддитивным белым и гауссовским, вероятность ошибки (т. е. вероятность перехода одного символа канала в другой) для всех сигналов одна и та же. Расстояние Хэмминга применяется также для описания процесса возникновения ошибок в запоминающих устройствах вычислительных машин, в которых разные двоичные символы слова хранятся в различных матрицах [5].

Для описания возникновения ошибок в системах с фазовой модуляцией может быть использовано *расстояние Ли* [6]. Чтобы ввести это расстояние, необходимо установить взаимно однозначное соответствие между элементами множества V_1 и целыми числами от 0 до $q-1$. Здесь для простоты будем считать, что $V_1 = \{0, 1, \ldots, q-1\}$. Для каждого элемента $a \in V_1$ определим число $|a|_L$ следующим образом: $|a|_L = a$, если $0 \leqslant a \leqslant (q-1)/2$, и $|a|_L = q - a$ в противном случае. Расстояние Ли $d_L(A, B)$ между последовательностями $A = (a_1, a_2, \ldots, a_n)$ и $B = (b_1, b_2, \ldots, b_n)$ определяется равенством $d_L(A, B) = \sum_{i=1}^{n} |a_i - b_i|_L$. При q, равном 2 или 3, расстояние Хэмминга совпадает с расстоянием Ли. Поскольку в теории кодирования результаты, полученные для расстояния Ли, не являются столь значительными, как результаты, полученные для расстояния Хэмминга, то в данной книге вопросы, связанные с расстоянием Ли, почти не рассматриваются (детальное изложение этих вопросов можно найти в книге Берлекэмпа [4]).

Для описания и исследования методов кодирования в системах с амплитудной модуляцией наиболее подходящим является евклидово расстояние (см. гл. 7), однако в теории кодирования получено лишь несколько частных результатов, посвященных этому вопросу, и они в основном связаны с некоторыми идеями устранения недостатков кодов, построенных для расстояния Хэмминга [7, 8].

Подмножество C множества V_n, состоящее из K последовательностей длины n, назовем q-ичным *блоковым кодом длины* n. Последовательности длины n, входящие в C, будем называть *кодовыми словами*. Число $R = (\log_q K)/n$ называется *скоростью передачи* кода C. Минимальное значение расстояния Хэмминга (расстояния Ли) между различными кодовыми словами кода C назовем минимальным расстоянием Хэмминга (минимальным расстоянием Ли) или просто *минимальным расстоянием* кода C.

Если $K = q^k$ $(1 \leqslant k \leqslant n)$ и в блоковом коде для любой последовательности $a_1, a_2, \ldots, a_k$ из k символов V_1 (символы в этой последовательности могут повторяться) найдется ровно одно кодовое слово, первые k компонент которого есть $a_1, a_2, \ldots, a_k$, то этот блоковый код называется *систематическим*. Большинство используемых на практике и обычно исследуемых кодов являются систематическими. *Кодер* систематического кода, на вход которого с выхода преобразователя «источник информации — q-ичная последовательность» поступает q-ичная последовательность символов множества V_1, разбивает входную последовательность на блоки длины k и для каждого такого блока $w_i = (a_1, a_2, \ldots, a_k)$ из k символов находит кодовое слово $\varphi_e(w_i)$, первые k символов которого совпадают с w_i (по определению существует ровно одно такое кодовое слово). Далее, к k символам $a_1, \ldots, a_k$ кодер приписывает в конце $n - k$ символов $a_{k+1}, \ldots, a_n$ и полученное таким образом кодовое слово посылает на выход как единый блок. Первые k компонент такого кодового слова называются *информационными символами*, а последние $n - k$ компонент — *проверочными символами*. При этом параметры k и $n - k$ называются соответственно *числом информационных* и *числом проверочных* (или *избыточных*) *символов*. Как мы видели в приведенных в предыдущем разделе примерах, избыточные $n - k$ символов могут использоваться для исправления или обнаружения ошибок. Отношение k/n обычно обозначают через R и называют *скоростью передачи*. Скорость передачи является параметром, характеризующим эффективность передачи.

Описанный выше кодер обрабатывает каждый поступающий блок независимо от других, так что каждое новое кодовое слово на его выходе оказывается не связанным с предыдущими кодовыми словами. Это является особенностью блоковых кодов. Кроме блоковых кодов, имеются другие типы кодов, в частности сверточные коды, которые детально рассматриваются в гл. 5 и 6. Каждый новый блок на выходе кодера сверточных кодов зависит от предыдущих блоков. Блоковые коды по сравнению с кодами других типов являются более простыми с точки зрения их математического описания, и, кроме того, они лучше исследованы.

Входные символы канала, которые являются в то же время выходными символами кодера, преобразуются в модуляторе в сигналы, которые могут быть переданы по каналу. Демодулятор выполняет обратную операцию, а именно каждому принятому сигналу, подвергшемуся воздействию шумов, он сопоставляет наиболее подходящий символ канала. Последовательность символов с выхода демодулятора поступает на вход декодера. *Декодер* блокового кода C разбивает эту последовательность на блоки в соответствии с разбиением последовательности на выходе кодера [1]) и далее для каждого принятого блока, используя свойства кода C, выбирает одно из наиболее вероятных кодовых слов, которые могли быть переданы. Это кодовое слово появляется на выходе декодера (в действительности на выходе декодера могут быть лишь информационные символы кодового слова). В некоторых случаях декодер может не принимать решения о том, какое кодовое слово передавалось, а послать на выход специальный символ D, означающий обнаружение ошибки. Таким образом, функция декодера заключается в реализации отображения φ_d множества V_n в множество $C \cup \{D\}$. Это отображение в принципе можно задать с помощью *таблицы декодирования*, которая каждой последовательности из V_n сопоставляет некоторое кодовое слово или символ D. В тех случаях, когда код C используется как *код, исправляющий ошибки, то* значениями $\varphi_d(w_r)$, $w_r \in V$, являются только кодовые слова. Если же код C используется как *код, обнаруживающий ошибки, то* $\varphi_d(w_r) = D$ для всех последовательностей w_r длины n, не являющихся кодовыми словами. Как видно из примера 1.1, существуют также коды, которые часть ошибок исправляют, а другие ошибки обнаруживают (вопросы целесообразности использования таких кодов будут рассмотрены в гл. 7).

В теории кодирования обычно предполагается, что кодер и демодулятор, точно так же, как и декодер и демодулятор, представляют собой отдельные устройства; при этом каждый символ канала преобразуется и передается независимо от других. Такое построение системы приводит не только к упрощению устройств, но, как мы увидим в гл. 7, и к некоторому снижению пропускной способности канала. Согласно теореме Шеннона для дискретного канала [9] (этот канал будет рассмотрен ниже в разд. 1.5), в случае использования подходящих (блоковых) кодов, исправляющих ошибки, при всех скоростях, меньших пропускной способности канала, информацию можно передавать со

---

[1]) Из-за воздействия шумов разбиение последовательности на блоки может оказаться неправильным. В таких случаях говорят, что произошла ошибка синхронизации. Для борьбы с ошибками синхронизации разработаны различные методы [2].

сколь угодно малой вероятностью ошибки. Однако теорема Шеннона является так называемой теоремой существования и не дает конкретных методов построения таких кодов. Разработка практически приемлемых методов кодирования является задачей теории кодирования. Для того чтобы эффективность использования кодов была высокой, необходимо усреднить воздействие шумов. Это приводит к необходимости выбирать длину кода достаточно большой, что в свою очередь обычно связано с чрезмерным усложнением кодеров и декодеров. Например, в случае $q = 2$, $R = 1/2$ и $n = 200$ число кодовых слов достигает приблизительно 10^{30}, а общее число последовательностей длины n — приблизительно 10^{60}, так что практически построить описанную таблицу декодирования просто невозможно. Этот пример показывает, что построение кодов, которые имели бы достаточно большую длину (а следовательно, и высокую эффективность), но в то же время были бы практически реализуемыми с точки зрения сложности кодера и декодера, является непростой задачей. Коды, удовлетворяющие этим условиям, должны иметь хорошую (алгебраическую) структуру, которая гарантировала бы нужные корректирующие способности и позволяла бы найти практически реализуемые процедуры кодирования и декодирования. Эта точка зрения на цели теории кодирования и пути их достижения была положена в основу исследований, которые велись и ведутся в настоящее время в области алгебраической теории кодирования.

Утверждение 1.5. *Если* $A_1, A_2, A_3 \in V_n$, *то* $d(A_1, A_3) \leqslant d(A_1, A_2) + d(A_2, A_3)$.

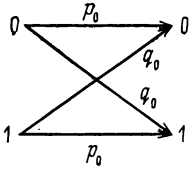
Утверждение 1.6. *Пусть* $D(A_1) = \{A \mid d(A_1, A) \leqslant t,\ A \in V_n\}$ *и* $D(A_2) = \{A \mid d(A_2, A) \leqslant t,\ A \in V_n\}$. *Тогда необходимым и достаточным условием того, что* $D(A_1)$ *и* $D(A_2)$ *не содержат общих элементов, является выполнение соотношения* $d(A_1, A_2) \geqslant 2t + 1$.

Краткое доказательство. 1) Допустим, что $d(A_1, A_2) \leqslant 2t$. Пусть $i_1, i_2, \ldots, i_{d(A_1, A_2)}$ — номера компонент, в которых различаются последовательности A_1 и A_2. Тогда если обозначить через A_3 последовательность, получающуюся из A_1 заменой компонент последней с номерами $i_1, i_2, \ldots, i_{[d(A_1, A_2)/2]}$ на соответствующие компоненты A_2, то $A_3 \in D(A_1)$ и $A_3 \in D(A_2)$.

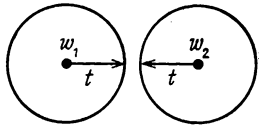
2) Пусть $d(A_1, A_2) \geqslant 2t + 1$. Предположим, что существует последовательность A_3, принадлежащая одновременно и $D(A_1)$ и $D(A_2)$. Тогда $d(A_1, A_3) \leqslant t$ и $d(A_2, A_3) \leqslant t$. Но в таком случае, как следует из утверждения 1.4, $d(A_1, A_2) \leqslant 2t$, что противоречит предположению.

1.3. Двоичный симметричный канал

Чтобы оценить эффективность использования кодов, необходимо знать статистические свойства используемого канала. При этом в состав канала входят также модулятор и демодулятор. Простейшей приближенной моделью канала является так называемый *двоичный симметричный канал* (ДСК). В ДСК принятый символ с вероятностью p_0 совпадает с переданным символом и с вероятностью $q_0 = 1 - p_0$, $p_0 > q_0$, отличается от последнего; при этом считается, что ошибки при передаче различных символов происходят независимо (фиг. 1.2). Такие каналы, в которых ошибки возникают независимо, называют каналами без памяти. Ошибки, возникающие в ДСК, называются случайными ошибками. Для ДСК вероятность того, что при передаче



Фиг. 1.2. Двоичный симметричный канал.



Фиг. 1.3. Расстояние Хэмминга и исправление ошибок.

некоторого кодового слова кода длины n ошибки возникнут в определенных i компонентах из n, равна $p_0^{n-i}q_0^i$. Поскольку $p_0 > q_0$, то чем меньше i, тем больше эта вероятность. Следовательно, если предположить, что все кодовые слова передаются с равными вероятностями, то наилучшим решением при приеме является такое, при котором переданным считается кодовое слово, отличающееся от принятой последовательности длины n в наименьшем числе компонент. Такое правило принятия решений называется *декодированием по максимуму правдоподобия*. Из вышеизложенного видно, что введенное в предыдущем разделе расстояние Хэмминга является понятием, важным для ДСК. Чтобы можно было исправлять ошибки в t и меньшем числе компонент (далее такие ошибки называются ошибками веса t и меньше или t-кратными ошибками), необходимо и достаточно, чтобы для любых двух различных кодовых слов w_1 и w_2 множество различных двоичных последовательностей длины n, отличающихся в t или меньшем числе компонент от w_1, т. е. $\{w \mid d(w, w_1) \leqslant t,\ w \in V_n\}$, и множество различных двоичных последовательностей длины n, отличающихся в t или меньшем числе компонент от w_2, т. е. $\{w \mid d(w, w_2) \leqslant t,\ w \in V_n\}$, не имели общих элементов (фиг. 1.3). Другими словами, как следует из утверждения 1.6, необходимо и достаточно, чтобы $d(w_1, w_2) \geqslant$

$\geqslant 2t+1$. Таким образом, код исправляет ошибки веса t и менее тогда и только тогда, когда минимальное расстояние кода больше или равно $2t+1$. Аналогично можно показать, что код обнаруживает любые ошибки веса t и менее (т. е. ошибки в t и меньшем числе компонент) тогда и только тогда, когда минимальное расстояние кода больше или равно $t+1$.

Утверждение 1.7. *Для того чтобы код исправлял любые ошибки веса* t_1 *и менее и обнаруживал любые ошибки веса от* t_1+1 *до* $t_2 (t_1 < t_2)$, *минимальное расстояние должно быть не меньше чем* t_1+t_2+1.

Краткое доказательство. Пусть w_1 и w_2 — два произвольных различных кодовых слова, а $D_{t_1}(w_1)=\{w \mid d(w, w_1) \leqslant t_1,\ w \in V_n\}$ и $D_{t_2}(w_2)=\{w \mid d(w, w_2) \leqslant t_2,\ w \in V_n\}$. Для того чтобы код удовлетворял указанному в утверждении 1.7 свойству, необходимо и достаточно, чтобы множества $D_{t_1}(w_1)$ и $D_{t_2}(w_2)$ не имели общих элементов.

Таким образом, как видно из вышеизложенного, минимальное расстояние d кода является важным параметром, показывающим, насколько хорошим является код в ДСК. Здесь следует заметить, что при $d=2t+1$ и $d=2t+2$ обычно удается исправлять также часть ошибок веса $t+1$ и более (см. пример 3.1 в разд. 3.2).

В некоторых случаях удается более или менее точно установить зависимость вероятности правильного декодирования от вероятности ошибки в канале, но полностью определить корректирующую способность кода только через минимальное расстояние кода d, вообще говоря, нельзя. Однако, поскольку, за исключением ряда случаев, когда значения параметров n и k очень малы, описать более детальные свойства кода почти невозможно, то обычно корректирующую способность кода характеризуют минимальным расстоянием d. Исходя из вышеизложенного, говорят, что: 1) среди кодов с одинаковыми параметрами n и k (т. е. в классе (n, k)-кодов) лучшим является код, который имеет большее минимальное расстояние d; 2) среди кодов с одинаковыми параметрами n и d лучшим является код, который имеет большее число информационных символов k, и 3) среди кодов с одинаковыми параметрами k и d лучшим является код, который имеет меньшую длину n, а следовательно, и меньшее число избыточных символов $r=n-k$. Заметим, что здесь при сравнении качества кодов не учитывают сложность кодирования и декодирования. Поэтому иногда говорят более точно: «код имеет лучшие параметры».

Существуют аналоги двоичного симметричного канала и в случае $q>2$. Для этих каналов, точно так же, как и для ДСК, можно получить необходимые и достаточные условия того, что

код исправляет (обнаруживает) ошибки веса t и менее (при выводе этих условий для ДСК мы не пользовались тем, что $q=2$). Однако при $q>2$ эта модель является адекватной моделью лишь тех каналов, в которых вероятность того, что переданный символ S будет ошибочно принят как некоторый символ S', отличный от S, является величиной, не зависящей ни от S, ни от S'.

Современные цифровые системы большей частью являются двоичными, поэтому двоичный случай является наиболее важным. Однако, поскольку, во-первых, многие алгебраические коды могут быть достаточно просто обобщены с двоичного случая на случай $q=p^m$ (p — простое число), во-вторых, из 2^l-ичных кодов могут быть получены некоторые превосходные двоичные коды [2], а в-третьих, и сами q-ичные коды ($q>2$) могут найти практическое применение, то в данной книге рассматривается и случай $q=p^m$ там, где это не усложняет изложение.

Корректирующая способность блоковых кодов, предназначенных для исправления случайных ошибок, не зависит от порядка следования компонент блока. В силу этого блоковые коды C и C' длины n, для которых существует такая перестановка Π упорядоченного множества $\{1, \ldots, n\}$, что $C'=\{(a_{\Pi(1)}, a_{\Pi(2)}, \ldots, a_{\Pi(n)}) \mid (a_1, a_2, \ldots, a_n) \in C\}$, называются *эквивалентными*.

ДСК, как и другие каналы без памяти, не дает адекватной модели некоторых реальных линий связи. В частности, в телефонных каналах и в запоминающих устройствах на магнитной ленте явно выражена тенденция к группированию ошибок [2]. При этом в среднем ошибки возникают сравнительно редко, однако в некоторые интервалы времени, когда шумы очень сильны, они появляются часто, а в другие интервалы, когда шумов почти нет, ошибки практически отсутствуют. Если ошибки возникли в пределах группы из b соседних компонент, то говорят, что возникла *пачка ошибок* длины b. Для коррекции таких пачек ошибок разрабатываются коды специальных типов (см. гл. 4).

1.4. Верхние границы для минимального расстояния кодов

Предельные возможности кодов, исправляющих ошибки, необходимо знать, во-первых, при оценке того, насколько реально используемые коды хуже «идеальных» кодов, во-вторых, при определении характеристик систем в целом и, в-третьих, для сравнения систем различных типов. В данном разделе будут получены простейшие верхние границы для минимального рас-

стояния [2, 4, 10]. Рассмотрение этих границ, кроме того, позволит ознакомить читателей с некоторыми принципиальными идеями теории информации.

Вначале получим *границу Чернова* [4], которая будет необходима в дальнейшем.

Лемма 1.1. *Пусть* $t < (q-1)n/q$. *Тогда*

$$\frac{q^{n\varphi(t/n)}}{n+1} \leqslant \binom{n}{t}(q-1)^t \leqslant \sum_{i=0}^{t}\binom{n}{i}(q-1)^i \leqslant q^{n\varphi(t/n)},$$

где $\varphi(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$.

Доказательство. Пусть $A(z) = (1+(q-1)z)^n = \sum_{i=0}^{n} A_i z^i$; ясно, что $A_i = \binom{n}{i}(q-1)^i$. Если $0 < z \leqslant 1$, то

$$\sum_{i=0}^{t}\binom{n}{i}(q-1)^i = \sum_{i=0}^{t} A_i \leqslant \sum_{i=0}^{t} A_i z^{i-t} = z^{-t}\sum_{i=0}^{t} A_i z^i \leqslant z^{-t}A(z). \quad (1.5)$$

С другой стороны, поскольку отношение $A_i/A_{i+1} = (i+1)/[(n-i)(q-1)]$ является монотонно возрастающей функцией i при $0 \leqslant i < n$ и, кроме того, $t < (q-1)n/q$, то найдется такое число $\bar{z}$, $0 < \bar{z} \leqslant 1$, что

$$\frac{A_{t-1}}{A_t} \leqslant \bar{z} \leqslant \frac{A_t}{A_{t+1}}.$$

Так как $A_{i-1}/A_i \leqslant A_{t-1}/A_t \leqslant \bar{z}$ при $0 \leqslant i < t$ и $\bar{z} \leqslant A_t/A_{t+1} \leqslant A_i/A_{i+1}$ при $t < i$, то для всех i

$$A_t\bar{z}^t \geqslant A_i\bar{z}^i,$$

и, следовательно, $A(\bar{z}) = \sum_{i=0}^{n} A_i\bar{z}^i \leqslant (n+1)A_t\bar{z}^t$. Отсюда и из формулы (1.5) получаем

$$\min_{0<z\leqslant 1}\frac{z^{-t}A(z)}{n+1} \leqslant A_t < \sum_{i=0}^{t} A_i \leqslant \min_{0<z\leqslant 1} z^{-t}A(z). \quad (1.6)$$

Положим $t/n = \tau$. Значение z, при котором достигается экстремум функции $z^{-n\tau}A(z) = z^{-n\tau}(1+(q-1)z)^n$, находится из уравнения

$$-n\tau z^{-n\tau-1}(1+(q-1)z)^n + nz^{-n\tau}(q-1)(1+(q-1)z)^{n-1} = 0.$$

Отсюда следует, что экстремум достигается в точке $z_e = \tau/[(1-\tau)(q-1)]$. Поскольку исследуемая функция является монотонно убывающей при $0 < z < z_e$ и монотонно

возрастающей при $z_e < z$, то в точке z_e она достигает минимума. Так как $t < (q-1)n/q$, то $z_e < 1$. В точке z_e рассматриваемая функция равна

$$\tau^{-n\tau}[(1-\tau)(q-1)]^{n\tau}(1-\tau)^{-n} = q^{n\varphi(\tau)}.$$

Отсюда и из формулы (1.6) получаем доказательство леммы 1.1.

Введенная выше функция $\varphi(x)$ при $q = 2$ совпадает с энтропией $H(x)$.

1.4.1. Верхняя граница Хэмминга

Здесь будет получена так называемая *верхняя граница Хэмминга* [3].

Теорема 1.1. *Если существует q-ичный блоковый код длины n со скоростью R и минимальным расстоянием Хэмминга $2t+1$ или более, то*

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leqslant q^{n(1-R)}.$$

Доказательство. Для каждого кодового слова w рассмотрим следующее множество $D(w)$ последовательностей длины n:

$$D(w) = \{w' \mid d(w, w') \leqslant t,\ w' \in V_n\},$$

т. е. $D(w)$ включает все последовательности w', отличающиеся от w не более чем в t компонентах. Следовательно, число последовательностей длины n, входящих в $D(w)$, равно

$$\sum_{i=0}^{t} \binom{n}{i} (q-1)^i,$$

Так как множества $D(w)$, соответствующие различным кодовым словам, не имеют общих элементов, то

$$K \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leqslant q^n,$$

где K — общее число кодовых слов. Отсюда и из равенства $K = q^{nR}$ следует справедливость теоремы.

Если для некоторого кода верхняя граница Хэмминга выполняется со знаком равенства, то код называется *совершенным*. Как можно видеть из вывода границы Хэмминга, для со-

вершенных кодов введенные выше множества $D(w)$, или, другими словами, «сферы» радиуса t, центрами которых являются кодовые слова, попарно не пересекаются и заполняют все пространство V_n. В силу этого совершенные коды иногда называют также плотноупакованными кодами.

Кроме тривиальных двоичных кодов из примера 1.1, кодовые слова которых получаются повторением одного и того же символа некоторое фиксированное число раз, совершенными кодами являются также следующие: 1) q-ичные совершенные коды, исправляющие одиночные ошибки и имеющие длину $n = (q^l - 1)/(q - 1)$, $q = p^m$, где p — простое число (*обобщенные коды Хэмминга* из утверждения 3.2 [3, 11] *нелинейные коды Васильева* [12] и *Шёнгейма* [13] с теми же параметрами), 2) *двоичный (23, 12)-код Голея, исправляющий тройные ошибки* (пример 4.2) и 3) *троичный (11,6)-код Голея, исправляющий двойные ошибки* (пример 4.3). Если q является степенью простого числа, то не существует других q-ичных совершенных кодов, за исключением кодов, эквивалентных указанным выше, и тривиальных нелинейных кодов, получающихся сложением каждого кодового слова указанных выше линейных совершенных кодов с некоторой фиксированной последовательностью длины n. Это было доказано Титвайненом [15][1]).

Утверждение 1.8. *Если существует q-ичный систематический совершенный код длины n, то*

$$n = (q^r - 1)/(q - 1),$$

где r — число проверочных символов.

Идея доказательства. Поскольку код систематический, то $R = k/n$. Далее следует воспользоваться верхней границей Хэмминга в частном случае $t = 1$.

Как будет показано ниже (утверждение 3.2), если q является степенью простого числа, q-ичные совершенные коды длины $n = (q^r - 1)/(q - 1)$ всегда существуют.

Утверждение 1.9. (Фарр [2, 4]). *Если q-ичный блоковый код длины $n = q^m - 1$ со скоростью $R \geqslant 1 - mt/n$ такой, что*

$$\frac{(t+1)!}{(q-1)^{t+1}} + \frac{(t+1)(t+2)}{2} \leqslant q^m,$$

то $d_{\text{мин}} \leqslant 2t + 2$.

---

[1]) В работе [26*] описаны все совершенные коды над алфавитами из $q = 2^k 3^l$ элементов. — *Прим. ред.*

Краткое доказательство.

$$\binom{q^m-1}{t+1}(q-1)^{t+1}=\frac{[(q-1)q^m]^{t+1}}{(t+1)!}\prod_{i=1}^{t+1}(1-iq^{-m})\geqslant$$

$$\geqslant\frac{[(q-1)q^m]^{t+1}}{(t+1)!}\left(1-\sum_{i=1}^{t+1}iq^{-m}\right)=$$

$$=\frac{[(q-1)q^m]^{t+1}}{(t+1)!}\left(1-\frac{q^{-m}(t+2)(t+1)}{2}\right)\geqslant$$

$$\geqslant q^{mt}\geqslant q^{n(1-R)}.$$

Отсюда и из теоремы 1.1 непосредственно следует, что неравенство $d_{\text{мин}}\geqslant 2t+3$ не может иметь места. Это утверждение используется для получения верхней оценки минимального расстояния некоторых двоичных БЧХ-кодов (см. гл. 4) [2, 4].

Верхнюю границу Хэмминга можно записать следующим образом:

$$R\leqslant 1-\frac{1}{n}\log_q\left[\sum_{i=0}^{t}\binom{n}{i}(q-1)^i\right].$$

Используя неравенство Чернова, можно показать, что при фиксированном отношении t/n и достаточно больших n граница Хэмминга имеет вид

$$R\leqslant 1-\varphi\left(\frac{t}{n}\right). \tag{1.7}$$

Так как $\varphi'(x)=\log_q(q-1)-\log_q x+\log_q(1-x)$, то в интервале $0\leqslant x\leqslant(q-1)/q$ функция $\varphi(x)$ монотонно возрастает от 0 до 1. Поэтому для любого R, $0<R<1$, существует единственное значение x, при котором $\varphi(x)=1-R$. Обозначим это значение x через $\delta(R)$. Из формулы (1.7) непосредственно следует справедливость следующего утверждения.

Следствие 1.1. *Для любого блокового кода со скоростью R и минимальным расстоянием $d_{\text{мин}}$ при достаточно больших n*

$$\frac{d_{\text{мин}}}{2n}\leqslant\delta(R). \tag{1.8}$$

1.4.2. Верхняя граница Плоткина

В области малых значений R верхняя граница Хэмминга является довольно грубой. При малых R более точной, чем граница Хэмминга, является верхняя граница Плоткина [16], определяемая следующей теоремой.

Теорема 1.2. *Если существует q-ичный блоковый код длины n с общим числом кодовых слов K и минимальным расстоянием* $d_{\text{мин}}$, *то*

$$d_{\text{мин}} \leqslant \frac{(q-1)\,nK}{q\,(K-1)}.$$

Доказательство. Эта граница получается путем оценки сверху среднего расстояния между кодовыми словами [4]. Пусть $w^{(1)}, w^{(2)}, \ldots, w^{(K)}$ — кодовые слова кода, существование которого предполагается в теореме, и пусть $w_m^{(i)}$ — m-я ($1 \leqslant m \leqslant n$) компонента кодового слова $w^{(i)}$. Сумма попарных расстояний Хэмминга между кодовыми словами определяется следующими равенствами:

$$D_t = \sum_{i=1}^{K} \sum_{j=1}^{K} d\left(w^{(i)}, w^{(j)}\right) = \sum_{i=1}^{K} \sum_{j=1}^{K} \sum_{m=1}^{n} d\left(w_m^{(i)}, w_m^{(j)}\right) =$$

$$= \sum_{m=1}^{n} \sum_{i=1}^{K} \sum_{j=1}^{K} d\left(w_m^{(i)}, w_m^{(j)}\right),$$

где функция $d\left(w_m^{(i)}, w_m^{(j)}\right)$ равна 1, если $w_m^{(i)} \neq w_m^{(j)}$, и равна 0, если $w_m^{(i)} = w_m^{(j)}$. Пусть $J_i^{(m)}$ — число равных i символов канала среди компонент $w_m^{(1)}, w_m^{(2)}, \ldots, w_m^{(K)}$. Тогда имеют место следующие равенства:

$$\sum_{i=1}^{q} J_i^{(m)} = K,$$

$$\sum_{i=1}^{K} \sum_{j=1}^{K} d\left(w_m^{(i)}, w_m^{(j)}\right) = \sum_{i=1}^{q} \sum_{j=1}^{q} \delta_{ij} J_i^{(m)} J_j^{(m)}, \tag{1.9}$$

где $\delta_{ij} = 1$, если $i \neq j$, и $\delta_{ij} = 0$, если $i = j$. Обозначим через Δ_q максимум квадратичной формы $\sum_{i=1}^{q} \sum_{j=1}^{q} \delta_{ij} x_i x_j$ при следующих условиях: $\sum_{i=1}^{q} x_i = K$ и $x_i \geqslant 0$ $(1 \leqslant i \leqslant m)$ (здесь будем считать, что переменные x_i принимают действительные значения). Методом индукции по q покажем, что $\Delta_q = K^2(q-1)/q$. В случае $q = 2$ справедливость последнего равенства проверяется легко. Полученные с использованием метода неопределенных множителей Лагранжа значения $x_1, \ldots, x_q$, при которых достигается экстремум рассматриваемой квадратичной формы, должны удовлетворять следующим условиям:

$$\sum_{j=1}^{q} \delta_{ij} x_j = \lambda, \quad 1 \leqslant i \leqslant q.$$

Отсюда следует, что $x_i = K/q$ $(1 \leqslant i \leqslant q)$. В этой точке значение квадратичной формы равно $K^2(q-1)/q$. Поскольку рассматриваемая область значений переменных x_i является ограниченной и замкнутой, то максимум квадратичной формы достигается либо в этой точке, либо на границе. На границе по крайней мере одна из переменных оказывается равной нулю, так что мы переходим к случаю, когда число переменных равно $q-1$ или меньше. По предположению индукции значения функции в таких точках не превосходят $K^2(q-2)/(q-1)$. Следовательно, $\Delta_q = K^2(q-1)/q$. С другой стороны, минимальное расстояние $d_{\text{мин}}$ должно удовлетворять неравенству

$$(K^2 - K)\, d_{\text{мин}} \leqslant D_t \leqslant nK^2(q-1)/q,$$

так как число пар различных кодовых слов равно $K^2 - K$. Теорема 1.2 доказана.

Аналогичная граница может быть получена и для расстояния Ли [4]. Из вывода границы Плоткина можно видеть, что кодами, минимальное расстояние которых достигает границы Плоткина, могут быть лишь коды, в которых расстояние между любыми двумя различными кодовыми словами одно и то же; такие коды называются *эквидистантными*. Примерами двоичных эквидистантных кодов являются *коды, состоящие из последовательностей максимальной длины* (см. пример 3.4), и коды, получающиеся из матриц Адамара.

1.4.3. Верхняя граница Элайса

Используя идеи доказательства верхних границ Хэмминга и Плоткина, Элайс получил новую границу для минимального расстояния, которая оказалась лучше обеих исходных границ по крайней мере при средних скоростях [2, 4]. Эта граница в дальнейшем получила название *границы Элайса* [1]). Рассмотрим q-ичный блоковый код C длины n с K кодовыми словами, скоростью R и минимальным расстоянием Хэмминга $d_{\text{мин}}$.

Предположим, что все последовательности длины n из V_n перенумерованы каким-либо образом числами от 1 до q^n. Пусть s — произвольное целое положительное число и K_i — число кодовых слов кода C, находящихся на расстоянии Хэмминга s или менее от i-й последовательности длины n. Поскольку каждое кодовое слово находится на расстоянии s или менее от

[1]) Этот результат получен Элайсом, доказательство было дано Бассалыго [27]. — *Прим. ред.*

$\sum_{i=0}^{s}\binom{n}{i}(q-1)^i$ последовательностей длины n, то

$$\sum_{i=1}^{q^n} K_i = KC_{n,q}(s), \quad \text{где} \quad C_{n,q}(s) = \sum_{i=0}^{s}\binom{n}{i}(q-1)^i.$$

Учитывая, что $K = q^{nR}$, получаем из последнего равенства следующую оценку снизу для $K' = \max\limits_i K_i$:

$$K' \geqslant KC_{n,q}(s)/q^n = C_{n,q}(s)/q^{n(1-R)}. \tag{1.10}$$

Предположим, что $K_j = K'$. Без ограничения общности можно считать, что $V_1 = \{0, 1, \ldots, q-1\}$. Пусть $(a_1, a_2, \ldots, a_n)$ — j-я последовательность длины n. Рассмотрим далее вместо блокового кода C код $C' = \{(b_1 - a_1), (b_2 - a_2), \ldots, (b_n - a_n) \mid (b_1, b_2, \ldots, b_n) \in C\}$, где знак — означает вычитание по модулю q. Если каждому кодовому слову $(b_1, \ldots, b_n)$ кода C сопоставить кодовое слово $(b_1 - a_1, b_2 - a_2, \ldots, b_n - a_n)$ кода C', то расстояние Хэмминга между кодовыми словами кода C будет совпадать с расстоянием Хэмминга между соответствующими словами кода C'. Следовательно, код C' имеет те же параметры n, K и $d_{\text{мин}}$, что и код C, и содержит K' кодовых слов, находящихся на расстоянии s или менее от последовательности $(0, 0, \ldots, 0)$ длины n. Поэтому вместо кода C с самого начала можно было рассматривать код C', так что мы просто будем считать, что указанные K' кодовых слов существуют в коде C. Обозначим через $J_i^{(m)}$ число таких кодовых слов среди K' кодовых слов кода C, находящихся на расстоянии s или менее от последовательности $(0, 0, \ldots, 0)$ длины n, у которых m-я компонента равна i. Но тогда, как и при определении границы Плоткина, получаем, что сумма D_t' попарных расстояний Хэмминга между указанными выше кодовыми словами определяется равенством

$$D_t' = \sum_{m=1}^{n}\sum_{i=0}^{q-1}\sum_{j=0}^{q-1}\delta_{ij}J_i^{(m)}J_j^{(m)},$$

где

$$K' = \sum_{i=0}^{q-1} J_i^{(m)}, \quad 1 \leqslant m \leqslant n.$$

Так как расстояние Хэмминга между любым из K' кодовых слов и кодовым словом $(0, 0, \ldots, 0)$ не превышает s, то

$$\sum_{m=1}^{n}\sum_{i=1}^{q-1} J_i^{(m)} \leqslant K's.$$

Поскольку D_t' и $d_{\text{мин}}$, как и раньше, связаны между собой неравенством

$$(K'^2 - K')\, d_{\text{мин}} \leqslant D_t', \tag{1.11}$$

то следует оценить сверху D_t'. Возьмем произвольное действительное число $y^{(m)}$, $0 \leqslant y^{(m)} \leqslant K'$, и найдем максимум $I = \sum_{i=0}^{q-1} \sum_{j=0}^{q-1} \delta_{ij} x_i x_j$ при следующих ограничениях:

$$\sum_{i=0}^{q-1} x_i = K', \tag{1.12}$$

$$\sum_{i=1}^{q-1} x_i = y^{(m)}, \tag{1.13}$$

$$x_i \leqslant 0, \quad 0 \leqslant i < q.$$

Поскольку, как следует из формул (1.12) и (1.13), $x_0 = K' - y^{(m)}$, то

$$I = \sum_{i=1}^{q-1} \sum_{j=1}^{q-1} \delta_{ij} x_i x_j + 2(K' - y^{(m)}) \sum_{i=1}^{q-1} x_i =$$

$$= \sum_{i=1}^{q-1} \sum_{j=1}^{q-1} \delta_{ij} x_i' x_j' - \frac{q-1}{q-2} (K' - y^{(m)})^2,$$

где $x_i' = x_i + (K' - y^{(m)})/(q-2)$. Ограничения на x_i' имеют следующий вид:

$$\sum_{i=1}^{q-1} x_i' = y^{(m)} + \frac{q-1}{q-2} (K' - y^{(m)}).$$

Таким образом, задача свелась к той же самой задаче нахождения максимума квадратичной формы, что решалась и при определении границы Плоткина. Искомый максимум $I(y^{(m)})$ квадратичной формы I определяется равенством

$$I(y^{(m)}) = \frac{[(q-1)K' - y^{(m)}]^2}{(q-2)(q-1)} - \frac{q-1}{q-2} (K' - y^{(m)})^2 =$$

$$= \left(2K' - \frac{q}{q-1} y^{(m)}\right) y^{(m)}.$$

Так как $\sum_{m=1}^{n} I(y^{(m)})$ является выпуклой вверх функцией $y^{(1)}, \ldots, y^{(n)}$, то с помощью метода множителей Лагранжа можно показать, что максимум $\sum_{m=1}^{n} I(y^{(m)})$ при условии $\sum_{m=1}^{n} y^{(m)} \leqslant K's$

достигается в точке

$$y^{(m)} = K's/n, \quad 1 \leqslant m \leqslant n,$$

и равен

$$K'^2 s\left(2 - \frac{qs}{(q-1)\,n}\right).$$

Отсюда и из неравенств (1.10) и (1.11) получаем

$$d_{\text{мин}} \leqslant \frac{K's}{K'-1}\left(2 - \frac{qs}{(q-1)\,n}\right) \leqslant \frac{sC_{n,q}(s)\,q^{-n(1-R)}}{C_{n,q}(s)\,q^{-n(1-R)} - 1}\left(2 - \frac{qs}{(q-1)\,n}\right).$$

Далее параметр s следует выбрать так, чтобы он минимизировал правую часть последнего неравенства. Однако, поскольку в общем случае это сделать сложно, рассмотрим здесь случай больших n. Из леммы 1.1 имеем

$$q^{n\varphi(s/n)}/(n+1) \leqslant C_{n,q}(s).$$

Если выбрать отношение s/n несколько большим, чем $\delta(R)$, и фиксировать, то при $n \to \infty$ произведение

$$C_{n,q}(s)\,q^{-n(1-R)}$$

будет стремиться к бесконечности, так что при больших n отношение $K'/(K'-1)$ можно считать равным единице. Таким образом, мы доказали следующую теорему.

Теорема 1.3. *Для любой скорости передачи R и любого положительного числа ε при достаточно большой длине кода n*

$$\frac{d_{\text{мин}}}{n} < \delta(R)\left(2 - \frac{q\delta(R)}{q-1}\right) + \varepsilon,$$

где $\delta(R)$ — функция, введенная выше в следствии 1.1.

Утверждение 1.10. *При фиксированном отношении t/n и достаточно больших n полученная выше верхняя граница лучше верхней границы (1.8).*

Утверждение 1.11. *При малых скоростях R и достаточно больших n полученная выше верхняя граница и граница Плоткина почти совпадают.*

Идея доказательства. Как указывалось выше (при введении функции $\delta(R)$), $\delta(R) \to (q-1)/q$ при $R \to 0$. С другой стороны, $K = q^{nR} \to \infty$ при $n \to \infty$.

В следующем разделе будет получена нижняя граница для вероятности ошибки, а в разд. 3.3 — нижняя граница Варшамова — Гилберта для минимального расстояния кодов. На

фиг. 3.1 приведены графики полученных выше верхних границ, а также нижней границы Варшамова — Гилберта при достаточно больших n [1]).

1.5. Теорема кодирования

Теория кодирования насчитывает уже более чем двадцатилетнюю историю и сейчас можно указать очень много примеров ее применения. Ученые и инженеры, которые работали в области теории кодирования на начальном этапе ее развития, считали основной своей задачей разработку практически приемлемых методов реализации теоремы кодирования Шеннона [9]. Согласно этой теореме, для широкого класса каналов при передаче информации со скоростью, меньшей пропускной способности канала, путем увеличения длины кода можно достичь сколь угодно малой вероятности ошибки при передаче. Известно, что в *дискретном канале без памяти* (предполагается, что шумы в канале являются случайными) вероятность ошибки при передаче p_e для кода длины n оценивается следующим образом:

$$e^{-n[E_{sp}\{R-O_1(n)\}+O_2(n)]} \leqslant p_e \leqslant Ae^{-nE(R)}, \qquad (1.14)$$

где $E(R)$ и $E_{sp}(R)$ — некоторые функции скорости передачи R и переходных вероятностей канала, $O_1(n)$ и $O_2(n)$ — функции, стремящиеся к нулю с ростом n, а A — константа. В области скоростей R, близких к пропускной способности канала, функции $E(R)$ и $E_{sp}(R)$ совпадают, т. е. $E_{sp}(R) = E(R)$. Как видно из неравенств (1.14), если значения функций $E(R)$ и $E_{sp}(R)$ положительны, то, выбрав длину кода n достаточно большой, вероятность ошибки можно сделать сколь угодно малой.

Верхняя граница вероятности ошибки в (1.14) была получена Шенноном путем определения средней вероятности ошибки в ансамбле кодов. Этот метод сравнительно сложен. Галлагер [10, 20] предложил более простой метод получения этой границы. Метод Галлагера оказался применимым не только для упрощения доказательства теоремы кодирования, он, в частности, широко используется при изучении поведения алгоритмов последовательного декодирования.

Функция $E_{sp}(R)$, входящая в нижнюю границу для вероятности ошибки в (1.14), была найдена Шенноном, Галлагером и Берлекэмпом методом, в основном аналогичным методу нахождения функции $E(R)$ [21].

---

[1]) Отметим, что в последнее время рядом авторов получены верхние границы минимального расстояния, улучшающие границу Элайса. — *Прим. ред.*

На основании вышеизложенного можно считать, что рассмотрение здесь метода Галлагера и других вопросов, связанных с верхней и нижней границами (1.14) для вероятности ошибки, будет полезным для понимания методов, которые оказали существенное влияние на развитие теории кодирования. С другой стороны, этот материал потребуется нам в дальнейшем.

1.5.1. Граница случайного кодирования

Поскольку верхняя граница для вероятности ошибки в (1.14) получается методом случайного кодирования, то она называется *границей случайного кодирования.*

Теорема 1.4. (Шеннон, Галлагер). *Пусть задан дискретный канал без памяти с входным алфавитом из K символов $u_1, u_2, \ldots, u_K$, выходным алфавитом из J символов $v_1, v_2, \ldots, v_J$ и переходными вероятностями $p_{jk} = p_r(v_j | u_k)$. Тогда для любой длины кода n, любого числа кодовых слов $M = e^{nR}$ и любого распределения вероятностей на множестве этих кодовых слов существует код длины n, вероятность ошибки для которого в указанном выше канале оценивается сверху следующим образом:*

$$p_e \leqslant \exp[-n\{-\nu R + E_0(\nu, \boldsymbol{Q})\}], \tag{1.15}$$

где

$$E_0(\nu, \boldsymbol{Q}) = -\ln \sum_{j=1}^{J} \left(\sum_{k=1}^{K} q_k p_{jk}^{1/(1+\nu)} \right)^{1+\nu}, \tag{1.16}$$

ν *— произвольное число из интервала* $0 \leqslant \nu \leqslant 1$, $\boldsymbol{Q} = (q_1, q_2, \ldots, q_k)$ *— произвольный вероятностный вектор* [1]).

Доказательство. Обозначим через U_n множество всех последовательностей длины n, которые могут быть переданы по каналу, а через V_n — множество последовательностей длины n, которые могут быть приняты на выходе канала. Пусть $p_r(\boldsymbol{V}|\boldsymbol{U})$ — вероятность того, что на выходе канала принято слово $\boldsymbol{V} \in V_n$ при условии, что было передано слово $\boldsymbol{U} \in V_n$.

Вначале рассмотрим некоторый фиксированный код, состоящий из M кодовых слов. Допустим, что целым числам $1, 2, \ldots, M$ сопоставлены соответственно кодовые слова $\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_m, \ldots, \boldsymbol{U}_M$, $\boldsymbol{U}_m \subset U_n$, $1 \leqslant m \leqslant M$. Предположим, что декодирование производится по максимуму правдоподобия.

---

[1]) Вероятностным вектором называется вектор $\mathbf{Q} = (q_1, \ldots, q_k)$, у которого $q_i \geqslant 0$, $\sum_{i=1}^{k} q_i = 1$. — *Прим. ред.*

А именно если принято слово V и целое число m таково, что при всех $m' \neq m$, $1 \leqslant m' \leqslant M$,

$$p_r(\boldsymbol{V} \mid \boldsymbol{U}_m) > p_r(\boldsymbol{V} \mid \boldsymbol{U}_{m'}), \tag{1.17}$$

то декодер максимального правдоподобия декодирует $\boldsymbol{V}$ в m. Если же принято слово $\boldsymbol{V}$, такое, что ни при каком m неравенства (1.17) одновременно для всех $m' \neq m$, $1 \leqslant m' \leqslant M$, не выполняются, то считается, что произошла ошибка декодирования. Вероятность возникновения ошибки декодирования при передаче m-го кодового слова обозначим через p_{em}. Эта вероятность определяется равенством

$$p_{em} = \sum_{V \in V_n} p_r(\boldsymbol{V} \mid \boldsymbol{U}_m) W_m(\boldsymbol{V}), \tag{1.18}$$

где

$$W_m(\boldsymbol{V}) = \begin{cases} 1, \text{ если } p_r(\boldsymbol{V} \mid \boldsymbol{U}_m) \leqslant p_r(\boldsymbol{V} \mid \boldsymbol{U}_{m'}) \text{ при некотором } m' \neq m, \\ 0 \text{ в остальных случаях.} \end{cases} \tag{1.19}$$

Первым шагом на пути вывода верхней границы для p_{em} является следующая оценка сверху для $W_m(\boldsymbol{V})$:

$$W_m(\boldsymbol{V}) \leqslant \left[\frac{\sum_{m' \neq m} p_r(\boldsymbol{V} \mid \boldsymbol{U}_{m'})^{1/(1+\nu)}}{p_r(\boldsymbol{V} \mid \boldsymbol{U}_m)^{1/(1+\nu)}} \right]^{\nu}, \quad \nu > 0. \tag{1.20}$$

В справедливости этого неравенства легко убедиться, если заметить, что

1) правая часть его всегда положительна, и, следовательно, оно выполняется всегда, когда $W_m(\boldsymbol{V}) = 0$;

2) в случае когда $W_m(\boldsymbol{V}) = 1$, одно из слагаемых в числителе правой части (1.20) больше знаменателя, а следовательно, и весь числитель больше знаменателя (при возведении числителя и знаменателя в ν-ю степень по-прежнему первый остается больше второго).

Подставив (1.20) в (1.18), получим

$$p_{em} \leqslant \sum_{V \in V_n} p_r(\boldsymbol{V} \mid \boldsymbol{U}_m)^{1/(1+\nu)} \left[\sum_{m' \neq m} p_r(\boldsymbol{V} \mid \boldsymbol{U}_{m'})^{1/(1+\nu)} \right]^{\nu}, \quad \nu > 0. \tag{1.21}$$

Это неравенство дает границу сверху для p_{em} в случае, когда множество кодовых слов задано, т. е. для фиксированного кода. Однако если число кодовых слов велико, то непосредственное вычисление границы (1.21) оказывается чрезвычайно затруднительным или вообще невозможным. В силу этого на следующем шаге осуществляется усреднение (1.21) по ансамблю кодов, выбранному некоторым специальным образом.

Зададим вначале на множестве U_n сообщений на входе канала распределение вероятностей $Q(U)$ и рассмотрим ансамбль кодов, в котором каждое кодовое слово выбирается независимо с распределением вероятностей $Q(U)$. А именно рассмотрим ансамбль кодов, в котором код, состоящий из M кодовых слов $U_1, U_2, \ldots, U_M$, имеет вероятность $\prod_{m=1}^{M} Q(U_m)$ [1]). Оценим сверху среднюю по этому ансамблю вероятность ошибки. Поскольку в любом ансамбле кодов всегда существует по крайней мере один код, вероятность ошибки для которого не больше, чем средняя по ансамблю вероятность ошибки, то таким образом можно доказать существование кода, для которого вероятность ошибки по крайней мере не больше, чем вероятность ошибки, средняя по ансамблю.

Итак, усредняя вероятность ошибки по ансамблю кодов, из формулы (1.21) получаем

$$\overline{p_{em}} \leqslant \sum_{V \in V_n} \overline{p_r(V \mid U_m)^{1/(1+\nu)} \left[\sum_{m' \neq m} p_r(V \mid U_{m'})^{1/(1+\nu)} \right]^{\nu}}, \quad (1.21')$$

где черта сверху означает усреднение. Ограничим область значений параметра ν интервалом $\nu \leqslant 1$ и следующим образом упростим формулу (1.21′):

1) Все слагаемые под чертой в формуле (1.21) являются случайными величинами. А именно каждое из слагаемых является функцией случайно выбираемых кодовых слов, принимающей действительные значения. Поскольку среднее значение суммы случайных величин равно сумме средних значений, то

$$\overline{p_{em}} \leqslant \sum_{V \in V_n} \overline{p_r(V \mid U_m)^{1/(1+\nu)}} \left[\sum_{m' \neq m} \overline{p_r(V \mid U_{m'})^{1/(1+\nu)}} \right]^{\nu}. \quad (1.21'')$$

2) Аналогично, поскольку среднее значение произведения независимых случайных величин равно произведению средних значений сомножителей, то из формулы (1.21″) получаем

$$\overline{p_{em}} \leqslant \sum_{V \in V_n} \overline{p_r(V \mid U_m)^{1/(1+\nu)}} \left[\sum_{m' \neq m} \overline{p_r(V \mid U_{m'})^{1/(1+\nu)}} \right]^{\nu}. \quad (1.21''')$$

3) Положим $\beta = \sum_{m' \neq m} p_r(V \mid U_{m'})^{1/(1+\nu)}$ и воспользуемся тем, что $\overline{\beta^{\nu}} \leqslant \overline{\beta}^{\nu}$ при $0 \leqslant \nu \leqslant 1$. В справедливости последнего неравенства легко убедиться, если заметить, что β^{ν} при всех $0 \leqslant \nu \leqslant 1$

[1]) Заметим, что среди кодовых слов $U_1, \ldots U_M$ могут быть и совпадающие. — *Прим. ред.*

является выпуклой вверх функцией аргумента β. Тогда

$$\overline{p_{em}} \leqslant \sum_{V \in V_n} \overline{p_r(V \mid U_m)^{1/(1+\nu)}} \Big[\sum_{m' \neq m} \overline{p_r(V \mid U_{m'})^{1/(1+\nu)}} \Big]^{\nu}. \quad (1.21'''')$$

4) Пользуясь опять тем, что среднее значение суммы случайных величин равно сумме средних значений слагаемых, получаем

$$\overline{\sum_{m' \neq m} p_r(V \mid U_{m'})^{1/(1+\nu)}} = \sum_{m' \neq m} \overline{p_r(V \mid U_{m'})^{1/(1+\nu)}}.$$

Отсюда и из формулы (1.21'''') следует, что

$$\overline{p_{em}} \leqslant \sum_{V \in V_n} \overline{p_r(V \mid U^m)^{1/(1+\nu)}} \Big[\sum_{m' \neq m} \overline{p_r(V \mid U_{m'})^{1/(1+\nu)}} \Big]^{\nu}. \quad (1.22)$$

Поскольку кодовые слова в рассматриваемом ансамбле кодов выбираются с распределением $Q(\boldsymbol{U})$, то

$$\overline{p_r(V \mid U_m)^{1/(1+\nu)}} = \sum_{U \in U_n} Q(\boldsymbol{U})\, p_r(V \mid \boldsymbol{U})^{1/(1+\nu)}. \quad (1.23)$$

Правая часть последнего равенства не зависит от m, и, следовательно, ее можно подставить в формулу (1.22) вместо соответствующих выражений как с индексом m, так и с индексом m'. Так как сумма по m' в (1.22) является суммой $M-1$ одинаковых слагаемых, то

$$\overline{p_{em}} \leqslant (M-1)^{\nu} \sum_{V \in V_n} \Big[\sum_{U \in U_n} Q(\boldsymbol{U})\, p_r(V \mid \boldsymbol{U})^{1/(1+\nu)} \Big]^{\nu+1}. \quad (1.24)$$

Оценка (1.24) справедлива для произвольного дискретного канала, задаваемого распределением $p_r(\boldsymbol{V} \mid \boldsymbol{U})$, а также произвольного распределения $Q(\boldsymbol{U})$ и любого $0 \leqslant \nu \leqslant 1$. Если предположить дополнительно, что канал является каналом без памяти, то оценку (1.24) можно упростить.

Пусть $u_1, u_2, \ldots, u_w, \ldots, u_n$ — компоненты входной последовательности $\boldsymbol{U}$, а $v_1, v_2, \ldots, v_w, \ldots, v_n$ — компоненты выходной последовательности $\boldsymbol{V}$. По определению в канале без памяти для любых $\boldsymbol{U} \in U_n$ и $\boldsymbol{V} \in V_n$

$$p_r(\boldsymbol{V} \mid \boldsymbol{U}) = \prod_{w=1}^{n} p_r(v_w \mid u_w). \quad (1.25)$$

Наложим на ансамбль кодов еще одно ограничение, а именно предположим, что символы кодовых слов выбираются незави-

симо с распределением $q(u)$, $u \in U_1$ (U_1 — входной алфавит канала). Тогда

$$Q(\boldsymbol{U}) = \prod_{w=1}^{n} q(u_w), \quad \boldsymbol{U} = (u_1, \ldots, u_w, \ldots, u_n). \tag{1.26}$$

Подставляя формулы (1.25) и (1.26) в (1.24), получаем

$$\overline{p_{em}} \leqslant (M-1)^{\nu} \sum_{V \in V_n} \left[\sum_{U \in V_n} \prod_{w=1}^{n} q(u_w)\, p_r(v_w \mid u_w)^{1/(1+\nu)} \right]^{1+\nu}. \tag{1.27}$$

Представляя далее сумму произведений в квадратных скобках в (1.27), как обычно, в виде произведения сумм, находим

$$\overline{p_{em}} \leqslant (M-1)^{\nu} \sum_{V \in V_n} \left[\prod_{w=1}^{n} \sum_{u_w \in U_1} q(u_w)\, p_r(v_w \mid u_w)^{1/(1+\nu)} \right]^{1+\nu}. \tag{1.28}$$

Аналогично, вынося знак произведения в формуле (1.28) за квадратные скобки, имеем

$$\overline{p_{em}} \leqslant (M-1)^{\nu} \prod_{w=1}^{n} \sum_{v_w \in V_1} \left[\sum_{u_w \in U_1} q(u_w)\, p_r(v_w \mid u_w)^{1/(1+\nu)} \right]^{1+\nu}, \tag{1.29}$$

$$0 \leqslant \nu \leqslant 1.$$

Правую часть этой формулы можно упростить следующим образом. Совокупность K входных символов $u_1, \ldots, u_k, \ldots, u_K$ обозначим через U_1, а совокупность J выходных символов $v_1, \ldots, v_j, \ldots, v_J$ — через V_1. Обозначим также через p_{jk} переходную вероятность канала $p_r(v_j \mid u_k)$, а через q_k — вероятность $q(u_k)$ выбора символа u_k в ансамбле кодов. Эти обозначения подставим в формулу (1.29) и, заметив, что все сомножители произведения в правой части (1.29) равны, получим

$$\overline{p_{em}} \leqslant (M-1)^{\nu} \left[\sum_{j=1}^{J} \left(\sum_{k=1}^{K} q_k p_{jk}^{1/(1+\nu)} \right)^{1+\nu} \right]^{n}, \quad 0 \leqslant \nu \leqslant 1. \tag{1.30}$$

Пусть R — скорость передачи информации в натуральных единицах (натах) на один символ канала (по определению $e^{nR} = M$). Если оценить сверху $M-1$ величиной $M = e^{nR}$, то соотношение (1.30) можно представить в виде

$$\overline{p_{em}} \leqslant \exp\left[-n \left\{ -\nu R - \ln \sum_{j=1}^{J} \left(\sum_{k=1}^{K} q_k p_{jk}^{1/(1+\nu)} \right)^{1+\nu} \right\} \right]. \tag{1.31}$$

Поскольку правая часть (1.31) не зависит от m, то она является также границей сверху для средней по ансамблю кодов вероятности ошибки декодирования. Так как в ансамбле кодов имеется по крайней мере один код, вероятность ошибки декодирования для которого не больше средней по ансамблю вероятности ошибки декодирования, то таким образом теорема 1.4 доказана.

Если провести минимизацию правой части формулы (1.31) по ν и $\boldsymbol{Q}$, то можно получить более точную оценку для p_e.

Следствие 1.2. *При выполнении условий теоремы 1.4 существует код, для которого*

$$p_e \leqslant \exp[-nE(R)], \tag{1.32}$$

$$E(R) = \max_{\nu, \boldsymbol{Q}} [-\nu R + E_0(\nu, \boldsymbol{Q})], \tag{1.33}$$

где максимум берется по всем ν из интервала $0 \leqslant \nu \leqslant 1$ и всем вероятностным векторам $\boldsymbol{Q}$.

Функция $E(R)$ называется функцией надежности, и ее свойства оказывают большое влияние на поведение верхней границы для p_e.

1.5.2. Свойства функции надежности

Нахождение максимума по ν и $\boldsymbol{Q}$ в формуле (1.33), естественно, зависит от того, как ведет себя функция $E_0(\nu, \boldsymbol{Q})$. Поэтому вначале исследуем свойства $E_0(\nu, \boldsymbol{Q})$ как функции ν и $\boldsymbol{Q}$.

Поведение $E_0(\nu, \boldsymbol{Q})$ как функции ν описывается следующей теоремой, доказательство которой было получено Галлагером [12].

Теорема 1.5. *Пусть задан канал с входным алфавитом из K символов, выходным алфавитом из J символов и переходными вероятностями p_{jk}, $1 \leqslant k \leqslant K$, $1 \leqslant j \leqslant J$. Будем считать, что вероятностный вектор $\boldsymbol{Q} = (q_1, \ldots, q_k)$ задает распределение вероятностей на множестве входных символов канала. Предположим, что средняя взаимная информация между входом и выходом канала*

$$I(\boldsymbol{Q}) = \sum_{k=1}^{K} \sum_{j=1}^{J} q_k p_{jk} \ln \frac{p_{jk}}{\sum_{i=1}^{K} q_i p_{ji}}$$

положительна. Тогда при $\nu \geqslant 0$ функция $E_0(\nu, \boldsymbol{Q})$ обладает следующими свойствами:

$$\text{(I)} \qquad E_0(\nu, \boldsymbol{Q}) = 0, \quad \nu = 0, \tag{1.34}$$

$$\text{(II)} \qquad E_0(\nu, \boldsymbol{Q}) > 0, \quad \nu > 0, \tag{1.35}$$

$$\text{(III)} \qquad \frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial \nu} > 0, \quad \nu > 0, \tag{1.36}$$

$$\text{(IV)} \qquad \left.\frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial \nu}\right|_{\nu=0} = I(\boldsymbol{Q}), \tag{1.37}$$

$$\text{(V)} \qquad \frac{\partial^2 E_0(\nu, \boldsymbol{Q})}{\partial \nu^2} \leqslant 0. \tag{1.38}$$

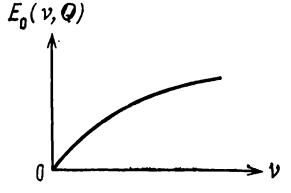
Равенство в (1.38) имеет место тогда и только тогда, когда при всех j и k, таких, что $q_k p_{jk} > 0$, имеет место равенство

$$\ln \frac{p_{jk}}{\sum\limits_k q_k p_{jk}} = I(\boldsymbol{Q}),$$

т. е. если случайная величина, являющаяся взаимной информацией, имеет нулевую дисперсию.

Доказательство опускается (см. работу [12]).

Соотношения (1.34)—(1.38) показывают, что $E_0(\nu, \boldsymbol{Q})$ как функция ν ведет себя так, как показано на фиг. 1.4.



Фиг. 1.4. Типичный график функции $E_0(\nu, \boldsymbol{Q})$.

Используя теорему 1.5, можно достаточно просто провести максимизацию $E(R)$ по ν при заданном $\boldsymbol{Q}$. Для этого введем функцию

$$E(R, \boldsymbol{Q}) = \max_{0 \leqslant \nu \leqslant 1} [-\nu R + E_0(\nu, \boldsymbol{Q})]. \tag{1.39}$$

Продифференцировав выражение в квадратных скобках в (1.39) по ν и приравняв его затем к нулю, получим

$$R = \frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial \nu}. \tag{1.40}$$

Если значение параметра ν, являющееся решением уравнения (1.40), лежит в интервале $0 \leqslant \nu \leqslant 1$, то оно максимизирует выражение (1.39). Так как, согласно формуле (1.38), правая часть (1.40) является невозрастающей функцией ν, то в силу равенства (1.37) уравнение (1.40) имеет решение всегда, когда

$$\left.\frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial \nu}\right|_{\nu=1} \leqslant R \leqslant I(\boldsymbol{Q}). \tag{1.41}$$

В этой области значений скорости передачи R функцию $E(R, \boldsymbol{Q})$ удобно задавать параметрически следующими соотношениями:

$$E(R, \boldsymbol{Q}) = E_0(\nu, \boldsymbol{Q}) - \nu \frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial \nu}, \tag{1.42}$$

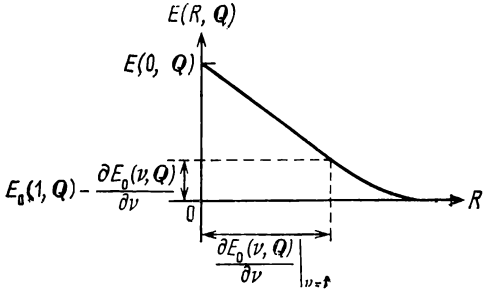
$$R = \frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial \nu}, \quad 0 \leqslant \nu \leqslant 1. \tag{1.43}$$

В случае когда $R < \partial E_0(\nu, \boldsymbol{Q})/\partial\nu\big|_{\nu=1}$, параметрическое задание $E(R, \boldsymbol{Q})$ посредством формул (1.42) и (1.43) не имеет смысла. В этой области скоростей передачи функция $-\nu R + E_0(\nu, R)$ возрастает с ростом ν, $0 \leqslant \nu \leqslant 1$, и достигает максимума при $\nu = 1$. Следовательно,

$$E(R, \boldsymbol{Q}) = E_0(1, \boldsymbol{Q}) - R, \quad R < \frac{\partial E_0(\nu, \boldsymbol{Q})}{\partial\nu}\bigg|_{\nu=1}. \tag{1.44}$$

На фиг. 1.5 показан график $E(R, \boldsymbol{Q})$ как функции R, построенный на основании формул (1.42)—(1.44).

Рассмотрим R и $E(R, \boldsymbol{Q})$ как функции ν, задаваемые формулами (1.42) и (1.43). С одной стороны, из соотношений (1.38), (1.40) и (1.43) можно видеть, что R является невозрастающей



Фиг. 1.5. Типичный график функции $E(R, \boldsymbol{Q})$

функцией ν. С другой стороны, дифференцируя выражение (1.42) по ν, получаем, что производная $E(R, \boldsymbol{Q})$ по ν равна -1, а следовательно, в силу (1.40) $E(R, \boldsymbol{Q})$ является неубывающей функцией ν при $\nu \geqslant 0$. Кроме того, заметим, что $E(R, \boldsymbol{Q}) = 0$ при $\nu = 0$. Отсюда приходим к выводу, что $E(R, \boldsymbol{Q}) > 0$ при всех $R < I(\boldsymbol{Q})$. При этом если выбрать в качестве $\boldsymbol{Q}$ распределение вероятностей, на котором достигается пропускная способность канала, то функция $E(R, \boldsymbol{Q})$ будет строго положительна при всех $R < C$. Это означает, что при всех $R < C$ вероятность ошибки убывает экспоненциально с ростом длины кодовых слов. Взяв отношение производных (1.42) и (1.43), получаем

$$\frac{\partial E(R, \boldsymbol{Q})}{\partial R} = -\nu, \tag{1.45}$$

т. е. параметр ν в формулах (1.42) и (1.43) можно интерпретировать как наклон графика функции $E(R, \boldsymbol{Q})$ к оси R.

Из замечания к неравенству (1.38), приведенному в теореме 1.5, видно, что $\partial^2 E_0(\nu, \boldsymbol{Q})/\partial\nu^2 = 0$ при всех $\nu \geqslant 0$, если только $\partial^2 E_0(\nu, \boldsymbol{Q})/\partial\nu^2 = 0$ при каком-либо одном значении $\nu \geqslant 0$. В этом случае связь между R и $E(R, \boldsymbol{Q})$ задается форму-

лой (1.44). Каналы, для которых $\partial^2 E_0(\nu, \boldsymbol{Q})/\partial\nu^2 = 0$, являются довольно искусственными и представляют собой каналы без шума. По этим каналам при всех скоростях, меньших пропускной способности, можно вести передачу с нулевой вероятностью ошибки.

Рассмотрим еще один способ максимизации $E(R, \boldsymbol{Q})$ по ν. Заметим, что при фиксированном ν функция $-\nu R + E_0(\nu, \boldsymbol{Q})$ является линейной по R, имеет наклон $-\nu$ и пересекается с осью ординат в точке $E_0(\nu, \boldsymbol{Q})$. Таким образом, $E_0(R, \boldsymbol{Q})$ является огибающей сверху семейства указанных выше прямых линий, соответствующих различным значениям параметра ρ из интервала [0, 1]. Отсюда следует, в частности, что $E(R, \boldsymbol{Q})$ является выпуклой вниз функцией R.

После того как проведена описанная выше максимизация по ν и в результате получена функция $E(R, \boldsymbol{Q})$, для нахождения функции $E(R)$ необходимо максимизировать $E(R, \boldsymbol{Q})$ по $\boldsymbol{Q}$, а именно

$$E(R) = \max_{\boldsymbol{Q}} E(R, \boldsymbol{Q}), \tag{1.46}$$

где максимум берется по всем K-мерным вероятностным векторам. Как видно из формулы (1.46), функция $E(R)$ является огибающей сверху семейства выпуклых вниз функций $E(R, \boldsymbol{Q})$. Это позволяет доказать следующую теорему.

Теорема 1.6. *Для любого дискретного канала без памяти функция надежности $E(R)$ является положительной непрерывной выпуклой вниз функцией R при $0 \leqslant R < C$.*

Доказательство. Если $C = 0$, то теорема тривиальна. Если $C \neq 0$, то при том $\boldsymbol{Q}$, на котором достигается пропускная способность канала, функция $E(R, \boldsymbol{Q})$ будет положительной при всех $0 \leqslant R < C$. Очевидно, что в этой области скоростей будет положительной и функция $E(R)$. Кроме того, как мы уже видели выше, при любом вероятностном векторе $\boldsymbol{Q}$ функция $E(R, \boldsymbol{Q})$ является непрерывной выпуклой вниз функцией R, производная которой изменяется в интервале от 0 до — 1. Отсюда следует, что и огибающая этого семейства кривых также является непрерывной выпуклой вниз функцией.

Далее рассмотрим, каким образом можно аналитически максимизировать $E(R, \boldsymbol{Q})$ по $\boldsymbol{Q}$. Соотношение (1.33), задающее $E(R)$, можно переписать в виде

$$E(R) = \max_{0 \leqslant \nu \leqslant 1} [-\nu R + \max_{\boldsymbol{Q}} E_0(\nu, \boldsymbol{Q})]. \tag{1.47}$$

Введем функцию

$$G(\nu, \boldsymbol{Q}) = \sum_j \left(\sum_k q_k p_{jk}^{1/(1+\nu)} \right)^{1+\nu}. \tag{1.48}$$

Тогда, как видно из формулы (1.16), $E_0(\nu, \boldsymbol{Q}) = -\ln G(\nu, \boldsymbol{Q})$, и, следовательно, минимизируя $G(\nu, \boldsymbol{Q})$ по $\boldsymbol{Q}$, можно максимизировать $E_0(\nu, \boldsymbol{Q})$ по $\boldsymbol{Q}$. Следующая теорема дает необходимые и достаточные условия, которым должен удовлетворять вектор $\boldsymbol{Q}$, минимизирующий $G(\nu, \boldsymbol{Q})$.

Теорема 1.7. *Функция $G(\nu, \boldsymbol{Q})$, задаваемая формулой (1.48), является выпуклой вниз функцией вероятностного вектора $\boldsymbol{Q} = (q_1, \ldots, q_K)$. Необходимыми и достаточными условиями того, что на вероятностном векторе $\boldsymbol{Q}$ достигается минимум $G(\nu, \boldsymbol{Q})$ [или максимум $E_0(\nu, \boldsymbol{Q})$], являются следующие неравенства:*

$$\sum_j p_{jk}^{1/(1+\nu)} \gamma_j^{\nu} \geqslant \sum_j \gamma_j^{1+\nu}, \tag{1.49}$$

где

$$\gamma_j = \sum_k q_k p_{jk}^{1/(1+\nu)}. \tag{1.50}$$

Эти неравенства при всех k, для которых $q_k \neq 0$, переходят в равенства.

Доказательство этой теоремы здесь опускается, однако его можно найти в работе Галлагера [20]. В случае когда все q_k положительны, соотношение (1.49) является простым следствием метода неопределенных коэффициентов Лагранжа максимизации $G(\nu, \boldsymbol{Q})$ при условии $\sum q_k = 1$. Неравенства (1.49) почти не облегчают задачу явного нахождения максимума $E_0(\nu, \boldsymbol{Q})$, но позволяют легко проверить, максимизирует ли тот или иной вероятностный вектор $\boldsymbol{Q}$ функцию $E_0(\nu, \boldsymbol{Q})$. Практически наиболее простыми методами максимизации $E_0(\nu, \boldsymbol{Q})$ оказываются численные методы.

После того как максимизация $E_0(\nu, \boldsymbol{Q})$ по $\boldsymbol{Q}$ проведена, функция $E(R)$ может быть найдена любым из указанных ниже трех способов:

1) Функция $E(R)$ может быть задана с помощью формулы (1.47) совокупностью прямых

$$-\nu R + \max_{\boldsymbol{Q}} E_0(\nu, \boldsymbol{Q}), \quad 0 \leqslant \nu < 1. \tag{1.51}$$

2) Построив график $\max_{\boldsymbol{Q}} E_0(\nu, \boldsymbol{Q})$ в зависимости от ν, $E(R)$ можно найти графически.

3) Функцию $E(R)$ можно найти также с помощью соотношений (1.42) и (1.43), взяв для каждого ν то значение $\boldsymbol{Q}$, на котором достигается максимум $E_0(\nu, \boldsymbol{Q})$.

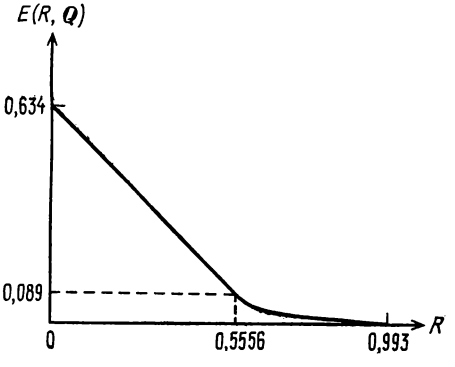
Пример 1.5. Рассмотрим двоичный симметричный канал с переходными вероятностями $p_{12} = p_{21} = p = 0{,}001$ и $p_{11} = p_{22} = 1 - p = 0{,}999$. Параметр $p = 0{,}001$ — это вероятность ошибки при передаче по каналу одного двоичного символа и

случае, если кодирование не производится. Очевидно, что в этом частном случае все компоненты вектора вероятностей $\boldsymbol{Q}$, максимизирующего $E_0(\nu, \boldsymbol{Q})$, равны $^1/_2$, а именно $q_1 = q_2 = {}^1/_2$. Для того чтобы убедиться в этом, достаточно подставить эти величины в формулу (1.49). Функция $E_0(\nu, \boldsymbol{Q})$ при таком значении $\boldsymbol{Q}$ имеет вид

$$E_0(\nu, \boldsymbol{Q}) = -\ln \sum_{j=1}^{2} \left(\sum_{k=1}^{2} q_k p_{jk}^{1/(1+\nu)} \right)^{1+\nu} =$$

$$= \nu \ln 2 - (1+\nu) \ln [p^{1/(1+\nu)} + (1-p)^{1/(1+\nu)}] =$$

$$= \nu \ln 2 - (1+\nu) \ln [0{,}001^{1/(1+\nu)} + 0{,}999^{1/(1+\nu)}]. \quad (1.52)$$



Фиг. 1.6. График функции $E(R, \boldsymbol{Q})$ для двоичного симметричного канала.

Дифференцируя (1.52) и используя параметрическое задание $E(R, \boldsymbol{Q})$, из соотношений (1.42) и (1.43) получаем

$$R = \ln 2 - H(p_\nu), \quad (1.53)$$

$$H(p_\nu) = -p_\nu \ln p_\nu - (1-p_\nu) \ln (1-p_\nu), \quad (1.54)$$

$$E(R, \boldsymbol{Q}) = p_\nu \ln \frac{p_\nu}{p} + (1-p_\nu) \ln \frac{1-p_\nu}{1-p}, \quad (1.55)$$

где

$$p_\nu = \frac{p^{1/(1+\nu)}}{p^{1/(1+\nu)} + (1-p)^{1/(1+\nu)}}. \quad (1.56)$$

Эти равенства справедливы при $0 \leqslant \nu \leqslant 1$, или, что то же самое, при

$$\ln 2 - H[\sqrt{p}/(\sqrt{p} + \sqrt{1-p})] = 0{,}5556 \leqslant R \leqslant C = 0{,}993.$$

При меньших скоростях $E(R, \boldsymbol{Q})$ находим по формуле (1.44)

$$E(R, \boldsymbol{Q}) = \ln 2 - 2 \ln (\sqrt{p} + \sqrt{1-p}) - R = 0{,}634 - R. \quad (1.57)$$

График функции $E(R, Q)$, задаваемой формулами (1.55) и (1.57), представлен на фиг. 1.6.

1.5.3. Граница сферической упаковки

Выше была получена верхняя граница для вероятности ошибки, и мы не рассматривали нижней границы вероятности ошибки в формуле (1.14). Вывод нижней границы, называемой границей сферической упаковки, несколько отличается от вывода верхней границы и детально рассмотрен в работах Галлагера [10] и Шеннона, Галлагера и Берлекэмпа [21]. Поэтому здесь мы только сформулируем результаты, относящиеся к нижней границе.

Теорема 1.8. (Граница сферической упаковки.) *Для любого кода длины n со скоростью передачи R в дискретном канале без памяти*

$$p_e \geqslant \exp[-n\{E_{sp}(R - O_1(n)) + O_2(n)\}], \tag{1.58}$$

где

$$E_{sp}(R) = \sup_{\nu > 0} [\max_{Q} E_0(\nu, Q) - \nu R] \tag{1.59}$$

и $E(\nu, Q)$ определяется формулой (1.16). Функции $O_1(n)$ и $O_2(n)$ стремятся к нулю с ростом n и определяются равенствами

$$O_1(n) = \frac{\ln 8}{n} + \frac{K \ln n}{n}, \tag{1.60}$$

$$O_2(n) = \frac{\ln 8}{n} + \sqrt{\frac{2}{n}} \ln \frac{e^2}{p_{\text{мин}}}, \tag{1.61}$$

где $p_{\text{мин}}$ — наименьшая ненулевая переходная вероятность канала, а K — число входных символов.

Доказательство опускается.

Как видно из формул (1.47), (1.51) и (1.59), величина $\max_{Q} E_0(\nu, Q)$ входит в показатели экспонент как верхней, так и нижней границ для вероятности ошибки и играет основную роль в поведении этих границ. Введем для этой величины специальное обозначение

$$E_0(\nu) = \max_{Q} E_0(\nu, Q). \tag{1.62}$$

Как известно из работы Шеннона, Галлагера и Берлекэмпа, теорему кодирования можно упростить, если ввести выпуклую вверх функцию $\hat{E}_0(\nu)$, примыкающую сверху к $E_0(\nu)$ [1]. Так,

[1]) То есть $\hat{E}_0(\nu) = \inf f(x)$, где нижняя грань берется по всем выпуклым вверх функциям f, для которых $f(\nu) \geqslant E_0(\nu)$ при всех $\nu \geqslant 0$. — *Прим. ред.*

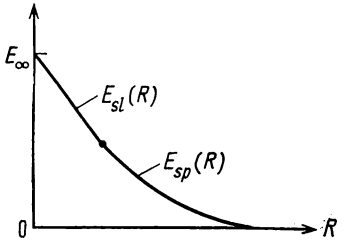
например, через $\hat{E}_0(\nu)$ функция $E_{sp}(R)$, задаваемая формулой (1.59), может быть выражена следующим образом:

$$E_{sp}(R) = \sup_{\nu>0} \{E_0(\nu) - \nu R\} = \sup_{\nu>0} \{\hat{E}_0(\nu) - \nu R\}. \qquad (1.63)$$

В действительности функция $\hat{E}_0(\nu)$ очень часто используется вместо $E_0(\nu)$, поскольку для большого числа каналов $E_0(\nu) = \hat{E}_0(\nu)$. Функцию $\hat{E}_0(\nu)$ называют внешним замыканием.

Граница сферической упаковки в области малых скоростей является неточной. В этой области скоростей следует пользоваться следующей прямолинейной границей.

Теорема 1.9. (Прямолинейная граница.) *Пусть задан произвольный дискретный канал без памяти, и пусть* $E_{sl}(R)$ *— линей-*



Фиг. 1.7. Типичные графики функций $E_{sp}(R)$ и $E_{sl}(R)$ для двоичного симметричного канала.

ная функция, которая касается кривой $E_{sp}(R)$ *и вместе с тем удовлетворяет условию* $E_{sl}(0) = E_\infty$, *где*

$$E_\infty = \max_{Q} - \sum_{i=1}^{K} \sum_{k=1}^{K} q_i q_k \ln \sum_{j=1}^{J} \sqrt{p_{ij} p_{kj}}. \qquad (1.64)$$

Пусть R_1 *— значение* R, *при котором* $E_{sl}(R) = E_{sp}(R)$. *Тогда для любого кода со скоростью* $R < R_1$ *вероятность ошибки* p_e *удовлетворяет неравенству*

$$p_e \geqslant 2 \exp[-n[E_{sl}\{R - O_3(n)\} + O_4(n)]], \qquad (1.65)$$

где $O_3(n)$ *и* $O_4(n)$ *— функции, стремящиеся к нулю с ростом* n.

Доказательство опускается.

Поведение кривых $E_{sp}(R)$ и $E_{sl}(R)$ для двоичного симметричного канала показано на фиг. 1.7.

1.5.4. Декодирование списком

В основе концепций, развиваемых в предыдущем разделе, лежит декодирование по максимуму правдоподобия, когда принятой последовательности символов сопоставляется одно из целых чисел. В данном разделе рассматривается более общая концепция декодирования, а именно декодирование списком. При декодировании списком принятой последовательности символов ставится в соответствие некоторый набор (список) целых чисел из интервала от 1 до M. Если целое число, соответствующее переданному сообщению, не входит в список целых чисел, полученный в результате декодирования, то говорят, что произошла ошибка при декодировании списком.

Допустим, что код и декодирование списком заданы. Пусть V_m — совокупность последовательностей длины n, которые могут быть приняты и которым сопоставляются списки сообщений, содержащие целое число m; пусть V_m^c — дополнение V_m. Тогда условная вероятность возникновения ошибки при передаче сообщения m и декодировании списком определяется равенством

$$p_{e,\,m} = \sum_{V \in V_m^c} p_r(\boldsymbol{V} \mid \boldsymbol{V}_m). \tag{1.66}$$

Полагая, что подлежащие передаче сообщения имеют равные вероятности и усредняя $p_{e,\,m}$ по m, можно ввести среднюю вероятность ошибки p_e для заданных кода и декодирования списком.

Пусть $p_e(n, M, L)$ — минимальная вероятность ошибки, которая может быть достигнута в заданном канале при декодировании списком из L сообщений и при использовании кодов длины n, содержащих M кодовых слов. В этих обозначениях $p_e(n, M, 1)$ — минимальная вероятность ошибки при обычном декодировании. Скорость передачи при декодировании списком определяется равенством

$$R = \frac{\ln (M/L)}{n} = \frac{\ln M}{n} - \frac{\ln L}{n}. \tag{1.67}$$

При $L = 1$ это определение совпадает с обычным определением скорости передачи.

Таким образом, декодирование списком является одним из методов принятия промежуточных «мягких» решений. Для декодирования списком Шенноном, Галлагером и Берлекэмпом [13] была получена следующая граница сферической упаковки.

Теорема 1.10. *В дискретном канале без памяти с переходными вероятностями* p_{kj}, $1 \leqslant k \leqslant K$, $1 \leqslant j \leqslant J$,

$$p_e(n, M, L) \geqslant \exp[-n[E_{sp}\{R - O_5(n)\} + O_6(n)]], \tag{1.68}$$

где

$$E_{sp}(R) = \sup_{\nu > 0} [E_0(\nu) - \nu R],$$

$$O_5(n) = \frac{\ln 8}{n} + \frac{K \ln n}{n}, \qquad (1.69)$$

$$O_6(n) = \sqrt{\frac{8}{n}} \ln \frac{e}{p_{\text{мин}}} + \frac{\ln 8}{n}.$$

Доказательство опускается.

Эта граница будет использована в дальнейшем при определении различных границ, характеризующих поведение алгоритмов последовательного декодирования.

Метод, основанный на построении границ для средних по некоторому ансамблю кодов характеристик и используемый при доказательстве теоремы кодирования и исследовании алгоритмов последовательного декодирования, является полезным по следующим причинам. Обычно с увеличением длины кода n число кодовых слов возрастает экспоненциально. При этом, как правило, ничего нельзя сказать о поведении характеристик каждого кода в отдельности. В то же время, вычислив средние по некоторому ансамблю кодов характеристики, можно утверждать, что характеристики по крайней мере одного из кодов этого ансамбля не хуже средних характеристик. Таким образом, метод исследования, основанный на вычислении границ для средних характеристик некоторого ансамбля кодов, оказывается полезным при анализе систем связи, характеристики которых зависят от очень большого числа случайных величин. В последние годы этот метод широко используется, кроме того, для получения оценок вероятности ошибки в системах передачи данных с межсимвольной интерференцией [22, 25].

2. Конечные поля

В данной главе рассматриваются теория групп, колец и полей, в частности конечных полей, в том объеме, который необходим для чтения литературы по теории кодирования. Для понимания материала этой главы необходимо знакомство с обозначениями, используемыми в теории множеств (но не с самой теорией множеств), а также с основными понятиями линейной алгебры. Основной текст и упражнения в этой главе одинаково важны. Упражнения могут быть примерами или задачами, могут входить в основной текст. Разделы, помеченные знаком *, при первом чтении могут быть опущены и прочитаны в дальнейшем, если в этом возникнет необходимость.

Запись $A \Rightarrow B$ означает, что из A следует B. Через Z обозначается множество целых чисел $\{0, \pm 1, \pm 2, \ldots\}$.

2.1. Группы

Важнейшими объектами, изучаемыми в алгебре, являются алгебраические системы. Примерами алгебраических систем являются группы, кольца, поля и др. Вообще говоря, под *алгебраическими системами* в алгебре понимаются множества, в которых определены одна или несколько *операций*, таких, например, как сложение или умножение. Операция в самом широком понимании — это закон, по которому некоторым упорядоченным парам элементов данного множества S ставятся в соответствие один или несколько элементов из S. Алгебраическую систему, образованную множеством S, в котором определены операции α, β, ..., будем обозначать через $\langle S, \alpha, \beta, \ldots\rangle$. Однако в тех случаях, когда ясно, каковы операции в системе, алгебраическая система $\langle S, \alpha, \beta, \ldots\rangle$ будет обозначаться просто через S.

Группы. Алгебраическая система $\langle G, \bigcirc\rangle$, образованная непустым множеством G и операцией $\bigcirc$, определенной для любых двух элементов a и b из G, называется *группой*, если выполнены следующие четыре аксиомы:

G. 1. Для любых двух элементов a и b из G однозначно определен элемент $a \bigcirc b \in G$.

G.2. Для любых трех элементов a, b и c из G

$$a \bigcirc (b \bigcirc c) = (a \bigcirc b) \bigcirc c.$$

G. 3. В G существует элемент e, называемый *единичным*, такой, что для любого элемента a из G

$$a \bigcirc e = e \bigcirc a = a.$$

G. 4. Для любого элемента a из G существует элемент x, называемый *обратным* к a, такой, что

$$x \bigcirc a = a \bigcirc x = e.$$

Если операция α такова, что для любых двух элементов a и b из S определен один элемент $a\alpha b \in S$, то говорят, что операция α *замкнута*. Таким образом, аксиома G. 1 постулирует замкнутость операции $\bigcirc$ в группе. Свойство операции, сформулированное в виде аксиомы G. 2, называется ассоциативностью. Ассоциативность операций означает, что последовательность выполнения ряда операций не влияет на результат. Аксиома G. 3 постулирует существование *единичного элемента*. Во множестве целых чисел 0 является единичным элементом по сложению, а 1 — по умножению. Аксиома G. 4 требует для каждого элемента множества G существования обратного элемента. Например, при сложении целых чисел обратным к числу a является $-a$.

Группа $\langle G, \bigcirc \rangle$ называется *коммутативной* или *абелевой*, если, кроме аксиом G. 1 — G. 4, справедлива также следующая аксиома коммутативности:

AG. 5. Для двух произвольных элементов a и b из G

$$a \bigcirc b = b \bigcirc a.$$

Если для обозначения групповой операции группы $\langle G, \bigcirc \rangle$, для которой выполняются аксиомы G. 1—G. 4 и AG. 5, вместо знака $\bigcirc$ используется знак $+$, то группа $\langle G, + \rangle$ называется по традиции *аддитивной группой*. Часто для обозначения групповой операции, вообще говоря некоммутативной, используется мультипликативная запись, т. е. вместо знака $\bigcirc$ используется знак $\cdot$, а вместо записи $a \bigcirc b$ — запись $a \cdot b$ или ab. При мультипликативной записи обратный элемент к a обозначается через a^{-1}. В результате того, что одни и те же стандартные обозначения, например знаки $+$ и $\cdot$, используются для разных операций в различных группах, возникает некоторый сознательно создаваемый беспорядок в использовании обозначений. С другой стороны, это можно рассматривать и как некоторый конструктивный способ расширения смысла знаков. Так, например, евклидово расстояние в двумерном случае определяется равенством $d(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$, $X = (x_1, x_2)$, $Y = (y_1, y_2)$, и обобщается на n-мерный случай следующим образом: $d(X, Y) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_n - y_n)^2}$. Однако заметим, что аналогично в гл. 1 были определены также расстояния

Хэмминга и Ли. В общем случае расстояние можно определить как функцию двух переменных, обладающую следующими тремя свойствами:

D. 1. $d(X, Y) = O \Leftrightarrow X = Y$,

D. 2. $d(X, Y) = d(Y, X)$,

D. 3. $d(X, Z) \leqslant d(X, Y) + d(Y, Z)$.

Понятие расстояния возникло в результате абстрагирования общих свойств «частных» расстояний. Однако после того, как расстояние определено указанным выше образом, все функции двух переменных, обладающие указанными выше тремя свойствами, могут быть названы единым термином — расстояние.

В дальнейшем будем придерживаться следующего правила в использовании знаков $+$ и $\cdot$. Будем считать что $+$ есть знак замкнутой операции, для которой $a + b = b + a$ (т. е. коммутативной операции), а $\cdot$ есть знак замкнутой операции, которая, вообще говоря, может не быть коммутативной.

Утверждение 2.1. *Алгебраическая система* $\langle Z, + \rangle$ *представляет собой группу, тогда как* $\langle Z, \cdot \rangle$ *группой не является (операции* $+$ *и* $\cdot$ *здесь являются соответственно обычными сложением и умножением целых чисел).*

Доказательство. Можно проверить, что $\langle Z, + \rangle$ удовлетворяет аксиомам G. 1—G. 4 и AG. 5. Следовательно, $\langle Z, + \rangle$ является группой, и, более того, коммутативной. Для алгебраической системы $\langle Z, \cdot \rangle$ аксиома G. 4 не выполняется, в частности, хотя бы потому, что уравнение $2 \cdot a = 1$ не имеет решения в множестве Z.

Утверждение 2.2. *Если в множестве* $S_3 = \{0, 1, 2\}$ *определить операцию* $+$ *с помощью табл.* 2.1 *(заметим, что эта операция коммутативна), то множество* S_3 *становится группой относительно введенной операции (другими словами,* $\langle S_3, + \rangle$ *есть группа).*

Таблица 2.1

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Этот пример интересен здесь по следующим причинам. Во-первых, он иллюстрирует использование знака $+$ для обозначения коммутативной операции. Во-вторых, операция $+$ не яв-

ляется обычным сложением. Так, в данном случае $2+2=1$, тогда как при обычном сложении $2+2=4$. Если бы операция $+$ была обычным сложением, то не выполнялась бы аксиома G. 1. Таким образом, введенная операция не является обычным сложением, но для нее выполняются аксиомы G. 1—G. 4 и AG. 5, и, как и при обычном сложении, справедливы следующие равенства:

$$2+(0+1)=(2+0)+1, \quad 2+1=1+2.$$

С помощью таблиц, аналогичных табл. 2.1, могут быть заданы и другие группы. Такие таблицы называются групповыми. Табл. 2.1 обладает следующими свойствами:

1) Она состоит лишь из элементов множества $\{0, 1, 2\}$.

2) Строки и столбцы ее являются перестановками последовательности $(0, 1, 2)$; все строки (столбцы) различны.

Аналогичными свойствами обладают все групповые таблицы.

Утверждение 2.3. *Множество комплексных чисел* $\{1, \omega, \omega^2\}$, *где* ω *— корень третьей степени из единицы, образует группу относительно обычной операции умножения комплексных чисел.*

Доказательство. Следует непосредственно проверить справедливость всех аксиом группы. Кроме того, доказательство можно легко получить, если воспользоваться групповой табл. 2.2.

Таблица 2.2

$\cdot$	1	ω	ω^2
1	1	ω	ω^2
ω	ω	ω^2	1
ω^2	ω^2	1	ω

Гомоморфизм и изоморфизм групп. Если отображение f множества S в S' таково, что для любого элемента $y \in S'$ существует элемент $x \in S$, такой, что $f(x) = y$, то говорят, что f *сюръективно*. Если отображение f таково, что $f(x_1) \neq f(x_2)$ для любых двух элементов x_1 и x_2 из S, то говорят, что f *инъективно*. Отображение f *биективно*, если оно одновременно сюръективно и инъективно [3]. Пусть даны две группы $\langle G, \alpha\rangle$ и $\langle G', \alpha'\rangle$. Если отображение f множества G в G' таково, что для любых двух элементов $a, b \in G$

$$f(a\alpha b) = f(a)\alpha' f(b),$$

то f называется *гомоморфным* или *гомоморфизмом* $\langle G, \alpha\rangle$ в $\langle G', \alpha'\rangle$. Далее, если, кроме того, отображение f множества G

в G' сюръективно или инъективно, то говорят, что гомоморфизм соответственно сюръективен или инъективен. Если гомоморфизм f одновременно и сюръективен и инъективен, то отображение f группы $\langle G, \alpha\rangle$ в $\langle G', \alpha'\rangle$ называется *изоморфным* или *изоморфизмом*. Две группы G и G' называются *изоморфными*, если существует изоморфное отображение G в G'. Две изоморфные группы идентичны с точки зрения их свойств и отличаются лишь формой представления элементов. С помощью биективного отображения f можно установить взаимно однозначное соответствие между элементами изоморфных групп так, что их групповые таблицы будут иметь одинаковый вид (с точностью до обозначения элементов). Группы, задаваемые табл. 2.1 и 2.2, изоморфны. Чтобы убедиться в этом, достаточно между элементами этих групп установить следующее взаимно однозначное соответствие:

$$0 \to 1 = \omega^0,$$
$$1 \to \omega = \omega^1,$$
$$2 \to \omega^2 = \omega^2.$$

Число элементов в группе G называется *порядком* G. Группа, порядок которой конечен, называется *конечной*. Если порядок группы бесконечен, то она называется *бесконечной*. Так как изоморфизм задает отношение эквивалентности, то все группы можно разбить на классы эквивалентности, каждый из которых можно было бы назвать абстрактной группой. При этом группы, принадлежащие одному классу эквивалентности и имеющие одинаковую «структуру», можно назвать конкретными группами. Определение числа классов эквивалентности групп заданного конечного порядка n является одной из нерешенных проблем теории конечных групп. В то же время, как это будет доказано ниже, все конечные поля, имеющие одинаковое число элементов, изоморфны. Структура всех конечных полей также хорошо известна. Здесь мы не будем глубоко вникать в проблемы теории конечных групп, однако для справок укажем число классов эквивалентности групп порядка $n \leqslant 10$:

Порядок групп n:	1	2	3	4	5	6	7	8	9	10
Число классов:	1	1	1	2	1	2	1	5	2	2

Задача 2.4. Показать, что все группы обладают следующими свойствами:

1. В группе содержится только один единичный элемент (единственность единичного элемента).

2. Для каждого элемента a группы обратный элемент определяется единственным образом (единственность обратного элемента).

3. Из $ab = ac$ следует, что $b = c$. Из $ba = ca$ также следует, что $b = c$.

4. Пусть a и b — два произвольных элемента группы. Тогда решениями уравнений $xa = b$ и $ay = b$ являются соответственно элементы $x = ba^{-1}$ и $y = a^{-1}b$, и эти решения единственны.

5. $(ab)^{-1} = b^{-1}a^{-1}$.

Решение. Предположим, что группа G содержит два единичных элемента e и e'. Тогда из аксиомы G. 3 имеем

$$ee' = e \quad \text{и} \quad ee' = e'.$$

Следовательно, $e = e'$. Таким образом, единичный элемент *единственный*. Доказательство свойств 2—5 предлагается провести читателю самостоятельно.

Подгруппы. Пусть $\langle G, \circ \rangle$ — группа, а H — подмножество G, являющееся группой относительно той же групповой операции $\circ$. Тогда H называется *подгруппой* группы G. Согласно этому определению, сама группа и группа $\langle \{e\}, \circ \rangle$, состоящая только из одного единичного элемента группы G, являются подгруппами группы G. Все подгруппы G, за исключением двух вышеуказанных (если они существуют), называются *собственными*. Например, множество E всех четных чисел является подмножеством множества Z, а $\langle E, + \rangle$ — собственная подгруппа группы $\langle Z, + \rangle$. Множество $\{0, 1, 2\}$ является подмножеством множества Z, но группа, задаваемая табл. 2.1, не является подгруппой группы $\langle Z, + \rangle$, так как эти группы определяются различными операциями.

Утверждение 2.5. *Необходимым и достаточным условием того, что непустое подмножество H элементов группы G является подгруппой группы G, является каждое из следующих соотношений:*

1) $a, b \in H \Rightarrow a \cdot b \in H$ *и* $a^{-1} \in H$;

2) $a, b \in H \Rightarrow ab^{-1} \in H$.

Пусть H — подгруппа группы G, а g — произвольный элемент из G. Множество $\{x \mid x = gh,\ h \in H\}$, обозначаемое ниже через gH, называется левым смежным классом группы G по подгруппе H, порождаемым элементом g. Аналогично множество $\{x \mid x = hg, h \in H\}$, обозначаемое через Hg, называется правым смежным классом группы G по подгруппе H, порождаемым элементом g.

Задача 2.6. Пусть $G = \langle \{0, 1, 2, 3, 4, 5\}, + \rangle$ — группа порядка 6, задаваемая табл. 2.3, а $H = \langle \{0, 3\}, + \rangle$ — подгруппа группы G, задаваемая табл. 2.4. Найти все правые смежные классы группы G по подгруппе H. Показать, что любые два левых смежных класса либо совпадают, либо не имеют общих элементов.

Кроме того, проверить, что объединение всех смежных классов совпадает с множеством G.

Таблица 2.3

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

Таблица 2.4

+	0	3
0	0	3
3	3	0

Решение. Поскольку в этой задаче используется аддитивная запись, то смежные классы также будем обозначать через $H+g$. Находя смежные классы, порожденные всеми элементами $g \in G$, получим

$$H+0=H+3=\{0, 3\}=H,$$
$$H+1=H+4=\{1, 4\},$$
$$H+2=H+5=\{2, 5\},$$
$$(H+0)\cap(H+1)=\phi,\ (H+0)\cap(H+2)=\phi,$$
$$(H+1)\cap(H+2)=\phi,\ (H+0)\cup(H+1)\cup(H+2)=G.$$

Отсюда непосредственно следует справедливость указанных выше свойств.

В приведенном выше примере каждый смежный класс содержит столько же элементов, сколько и подгруппа H. Это свойство присуще любой конечной группе G и любой ее подгруппе H, а именно для произвольного смежного класса Hg группы G по подгруппе H имеем

$$|Hg|=|H|. \tag{2.1}$$

Здесь и в дальнейшем через $|S|$ обозначается число элементов в множестве S. Равенство (2.1) доказывается следующим

образом. Пусть $H = \{h_1, \ldots, h_n\}$. Тогда, пользуясь определением Hg, можно установить следующее соответствие между элементами H и Hg:

$$H : h_1 \quad h_2 \quad \ldots \quad h_n,$$
$$Hg : h_1 g \quad h_2 g \quad \ldots \quad h_n g.$$

Отсюда следует, что $|Hg| \leqslant |H|$. Предположим, что $|Hg| < |H|$. Тогда найдутся два различных элемента h_i и h_j, таких, что $h_i g = h_j g$. В силу свойства 3 группы, указанного в задаче 2.4, получим $h_i = h_j$, что противоречит предположению о том, что элементы h_i и h_j различны. Следовательно, справедливость равенства (2.1) доказана. В приведенном выше упражнении $H + 0 = H + 3$ и т. д. Эти равенства являются следствием следующего общего свойства групп. Если смежные классы Ha и Hb группы G по подгруппе H имеют хотя бы один общий элемент, то они совпадают, т. е. $Ha = Hb$. В самом деле, если два смежных класса Ha и Hb имеют хотя бы один общий элемент, то $h_1 a = h_2 b$, где $h_1, h_2 \in H$. Умножая обе части последнего равенства слева на h_1^{-1}, получим $a = h_1^{-1} h_2 b$, где $h_1^{-1} h_2 \in H$. Тогда

$$Ha = \{ha \mid h \in H\} = \{h(h_1^{-1} h_2 b) \mid h \in H\} = \{h'b \mid h' \in H\} \subset Hb,$$

где $h' = hh_1^{-1} h_2 \in H$. Точно так же можно доказать, что $Hb \subset Ha$. Следовательно,

$$Ha = Hb.$$

Отсюда следует справедливость следующей теоремы Лагранжа.

Теорема 2.1. (Лагранж.) *Порядок любой подгруппы конечной группы является делителем порядка группы.*

Доказательство. Пусть H — произвольная подгруппа группы G. Обозначим через m и n соответственно порядки G и H. Если $m = n$, то $G = H$ и теорема доказана. Если $m > n$, возьмем некоторый элемент $a_1 \in G$, не входящий в H, и построим правый смежный класс Ha_1. Общее число элементов, входящих в H и Ha_1, равно $2n$. Если $m > 2n$, возьмем некоторый элемент $a_2 \in G$, такой, что $a_2 \notin H \cup Ha_1$, и построим смежный класс Ha_2. Число элементов в $H \cup Ha_1 \cup Ha_2$ равно $3n$. Продолжим эту процедуру до тех пор, пока все элементы G не будут включены либо в H, либо в один из смежных классов группы G по подгруппе H. Если число смежных классов, включая H, равно l, то $m = ln$.

Число левых смежных классов группы G по подгруппе H равно числу правых смежных классов. Это число обозначается через $[G : H]$ и называется индексом подгруппы H в группе G.

Из теоремы Лагранжа непосредственно следует справедливость следующего утверждения.

Следствие 2.1. *Если G — конечная группа, то*

$$|G| = [G : H]\,|H|.$$

Задача 2.7. Доказать, что группа G, которая рассматривалась в предыдущей задаче, может содержать подгруппы порядка 1, 2, 3 и 6, но не 4 и 5.

Решение. Порядок G равен 6, и все его делители — это числа 1, 2, 3 и 6. Подгруппой порядка 1 является подгруппа $\{0\}$, а подгруппой порядка 6 — сама группа G.

Нормальные делители. Подгруппа H группы G называется *нормальным делителем*, если для любого элемента $g \in G$

$$Hg = gH \quad \text{или} \quad gHg^{-1} = H = g^{-1}Hg.$$

Если H — подгруппа группы G, то, вообще говоря, левый смежный класс группы G по подгруппе H не совпадает ни с каким правым смежным классом. Однако если H — нормальный делитель, то, как следует из вышеприведенного определения, $Hg = gH$ и, следовательно, необходимость в различении левых и правых смежных классов отпадает. В этом случае можно говорить просто о смежном классе, порождаемом элементом g. В коммутативных группах все подгруппы являются нормальными делителями.

Факторгруппа. Пусть подгруппа H группы G является нормальным делителем. *Факторгруппа* G/H группы по нормальному делителю H определяется следующим образом. Обозначим через Hg смежный класс группы G по подгруппе H, содержащий элемент g, а именно $Hg = \{hg \mid h \in H\}$. Конечно, такое представление смежных классов не однозначно в том смысле, что для представления одного и того же смежного класса могут использоваться различные элементы g. Введем во множестве смежных классов операцию $\cdot$ следующим образом:

$$Hg_i \cdot Hg_j = \{g'g'' \mid g' \in Hg_i,\ g'' \in Hg_j\}. \tag{2.2}$$

Факторгруппа группы G по нормальному делителю H определяется как алгебраическая система, элементами которой являются смежные классы группы G по нормальному делителю H с операцией $\cdot$, задаваемой соотношением (2.2). Как будет показано ниже, эта алгебраическая система является группой.

Прежде всего покажем, что для этой алгебраической системы выполняется аксиома G. 1. Для этого вначале покажем, что множество $Hg_i \cdot Hg_j$, определяемое соотношением (2.2), является одним из смежных классов группы G по нормальному дели-

телю H. В самом деле,

$$Hg_i \cdot Hg_j = \{g'g'' \mid g' \in Hg_i = g_iH,\ g'' \in Hg_j\} =$$
$$= \{g_ih_1h_2g_j \mid h_1,\ h_2 \in H\} = \{g_ihg_j \mid h \in H\} =$$
$$= \{g_ig_j(g_j^{-1}hg_j) \mid h \in H\}.$$

Так как H является нормальным делителем, то $\{g_j^{-1}hg_j \mid h \in H\} = H$. Следовательно,

$$Hg_i \cdot Hg_j = \{g_ig_jh \mid h \in H\} = Hg_k,$$

где $g_k = g_ig_j$. Таким образом, справедливость аксиомы G. 1 доказана. Используя вышеприведенные результаты, можно показать справедливость аксиомы G. 2. Действительно,

$$(Hg_1 \cdot Hg_2) \cdot Hg_3 = Hg_1g_2 \cdot Hg_3 = Hg_1g_2g_3 =$$
$$= Hg_1 \cdot Hg_2g_3 = Hg_1 \cdot (Hg_2 \cdot Hg_3).$$

Точно так же можно показать, что сама подгруппа G выполняет роль единичного элемента, а элемент Hg^{-1} является обратным к элементу Hg. Таким образом, G/H удовлетворяет всем аксиомам группы.

Утверждение 2.8. *Пусть $G = \langle Z, + \rangle$. Обозначим через (n) множество кратных целого положительного числа n, а именно $(n) = \{nz \mid z \in Z\}$. Так как группа G коммутативна, то алгебраическая система $H = \langle (n), + \rangle$ является нормальным делителем группы G. Следовательно, можно определить факторгруппу G/H. Эту факторгруппу часто обозначают через Z_n.*

Доказательство. То, что H является подгруппой группы G, проверяется легко. Рассмотрим несколько более детально группу Z_n. Обозначим через $[x]$ смежный класс, содержащий x. Всего имеется n следующих смежных классов:

$$[0] = \{0,\ \pm n,\ \pm 2n,\ \ldots\} = (n),$$
$$[1] = \{1,\ 1 \pm n,\ 1 \pm 2n,\ \ldots\},$$
$$\vdots$$
$$[n-1] = \{n-1,\ n-1 \pm n,\ n-1 \pm 2n,\ \ldots\}.$$

Элемент, используемый для представления того или иного смежного класса, называется представителем этого смежного класса. В данном случае в качестве представителей смежных классов используются минимальные неотрицательные целые числа, входящие в соответствующие смежные классы. Если ввести операцию сложения смежных классов с помощью соотношения

$[i]+[j]=[i+j]$, то эти смежные классы будут образовывать группу.

Групповую таблицу группы Z_6 можно получить из табл. 2.3, если заменить в последней элементы i на $[i]$.

Циклические подгруппы. Рассмотрим один элемент a группы G. По примеру обычных экспоненциальных выражений элементы $a \cdot a$, $a \cdot a \cdot a$, ... будем обозначать соответственно через a^2, a^3, ... (при использовании аддитивной записи эти же элементы обозначаются через $2a$, $3a$, ...). Предположим, что группа G конечна, и рассмотрим следующую последовательность элементов:

$$a,\ a^2,\ a^3,\ \ldots. \tag{2.3}$$

Так как группа G конечна, то существуют такие числа i и j, что

$$a^i = a^j \qquad (i < j). \tag{2.4}$$

Но тогда, поскольку

$$a^i = a^j = a^i \cdot a^{j-i},$$

$a^{j-i} = e$. Минимальное целое положительное число m, такое, что $a^m = e$, называют *порядком* элемента a. Если группа G конечна, то порядок каждого элемента группы G также конечен. Однако в бесконечных группах порядок каждого элемента группы не обязательно должен быть конечным. Например, рассмотрим группу $\langle Z, + \rangle$. Для элемента 1 этой группы последовательность (2.3) имеет следующий вид:

$$1,\ 2,\ 3,\ 4,\ \ldots.$$

В этом случае равенство $m \cdot 1 = m = 0$ не выполняется ни при каком целом положительном m. В подобных случаях говорят, что порядок элемента бесконечен.

Задача 2.9. Определить порядок каждого элемента группы G из задачи 2.6.

Решение.

0:	$0=0$	Порядок 1
1:	1, 2, 3, 4, 5, 0	6
2:	2, 4, 0	3
3:	3, 0	2
4:	4, 2, 0	3
5:	5, 4, 3, 2, 1, 0	6

В данном случае порядок каждого элемента определялся по числу различных элементов в соответствующей последовательности (2.3). Хотя по определению порядок элемента — это ми-

нимальная разность $j - i$ чисел j и i, удовлетворяющих (2.4), использованный здесь метод нахождения порядка приводит к правильным результатам.

Наконец, заметим, что если m есть порядок элемента a, то все m элементов $a, a_2, \ldots, a^m = e$ различны. В противном случае мы имели бы $a^i = a^j$ $(1 \leqslant i < j \leqslant m)$ и, поскольку $a^i = a^i \cdot a^{j-i}$, получили бы $a^{j-i} = e$, где $j - i < m$. Это противоречит определению m.

Теорема 2.2. *Пусть a — элемент порядка m группы G. Тогда $H = \{a, a^2, \ldots, a^m\} = \{e, a, a^2, \ldots, a^{m-1}\}$ — подгруппа группы G.*

Доказательство. В H содержится единичный элемент e. Очевидно, что в H выполняется также аксиома ассоциативности. Кроме того, поскольку равенство $a^i \cdot a^{m-i} = a^m = e$ справедливо при всех $0 \leqslant i \leqslant m$, то для любого элемента $a^i \in H$ в H существует обратный элемент. Пусть a^i и a^j — два произвольных элемента из H. Если $i + j \leqslant m$, то $a^i \cdot a^j = a^{i+j} \in H$. Если $i + j > m$, то $a^i \cdot a^j = a^{i+j-m} \cdot a^m = a^{i+j-m}(i + j - m < m)$, а это вновь означает, что $a^i \cdot a^j \in H$. Таким образом, в H выполняется и аксиома G. 1, а следовательно, H есть группа.

Рассмотренная выше подгруппа H называется *циклической подгруппой*. В частности, если в группе G существует элемент a, такой, что $G = \{e, a, a^2, \ldots, a^{n-1}\}$, и $n = |G|$ $(a^n = e)$, то сама группа G называется циклической группой, порожденной элементом a.

Заметим, что слово порядок употребляется в следующих сочетаниях: порядок группы и порядок элемента группы, причем порядок элемента группы равен порядку циклической подгруппы, порожденной данным элементом. Из теорем 2.1 и 2.2 вытекает справедливость следующего утверждения.

Следствие 2.2. *Порядок каждого элемента конечной группы G является делителем порядка G.*

Утверждение 2.10. *Для произвольного целого числа p существует циклическая группа порядка p.*

Доказательство. Рассмотрим группу Z_n из утверждения 2.8. Легко убедиться в том, что Z_n порождается элементом [1], а следовательно, является циклической. Группа, рассмотренная в задаче 2.6, есть не что иное, как частный случай группы Z_n при $n = 6$. Заметим, что элемент 1 является не единственным элементом, порождающим эту группу. В частности, группа Z_6 может быть порождена также элементом 5 (см. задачу 2.9). Такие элементы, порождающие группу, называются порождающими элементами.

Утверждение 2.11. *Все циклические группы одного и того же порядка $m(<\infty)$ изоморфны.*

Доказательство. Пусть $G = \{a, a^2, \ldots, a^{m-1}, e\}$ и $H = \{b, b^2, \ldots, b^{m-1}, e\}$ — две циклические группы порядка m. Введем следующее отображение G в H:

$$f(a^i) = b^i, \quad i = 1, \ldots, m.$$

Очевидно, что отображение f биективно. Поскольку для любых двух элементов a^i и a^j из G

$$f(a^i \cdot a^j) = \begin{cases} f(a^{i+j}) = b^{i+j} = b^i b^j, & \text{если } i+j \leqslant m, \\ f(a^{i+j-m} \cdot a^m) = f(a^{i+j-m}) = b^{i+j-m} = b^i b^j, & \text{если } i+j > m, \end{cases}$$

то f является гомоморфным отображением G в H. Следовательно, G и H изоморфны.

Вообще говоря, две конечные группы одного порядка не обязательно должны быть изоморфными (см. задачу 2.4). Однако если ограничиться только циклическими группами, то, как мы видели выше, из равенства порядков групп следует их изоморфизм. Если группы G и H изоморфны, обычно одну из этих групп, например H, называют *представлением* G. Так, группа Z_p, введенная в утверждении 2.8, является одним из представлений циклических групп порядка p. Обычно удается получить несколько представлений групп, и выбор используемого представления определяется соображениями удобства.

Утверждение 2.12. *Все конечные группы, порядок которых является простым числом, циклические.*

Доказательство следует из утверждения 2.2.

Утверждение 2.13. *Пусть m — порядок циклической группы, порожденной элементом a. Тогда порядок элемента a^i равен $m/(m, i)$, где (m, i) — наибольший общий делитель чисел m и i.*

Доказательство. Используемая ниже запись $y|x$ означает, что y делит x. Пусть l — порядок элемента a^i. По определению l является минимальным целым числом, таким, что $a^{il} = e$. Отсюда и из того, что порядок a равен m, следует, что l — минимальное целое число, такое, что $m|il$. Это означает, что $l = m/(m, i)$.

Утверждение 2.14. *Пусть m и n — соответственно порядки элементов a и b группы G. Если $ab = ba$ и $n \nmid m$, то в G существует элемент порядка $mn/(m, n)$.*

Доказательство. Предположим сначала, что $(m, n) = 1$, т. е. что числа m и n взаимно простые. В этом случае элемент ab является искомым. В самом деле, поскольку $(ab)^{mn} = a^{mn}b^{mn} = e$, то порядок l элемента ab делит mn, т. е. $l \mid mn$. С другой стороны, поскольку $e = (ab)^{lm} = a^{lm}b^{lm} = b^{lm}$, то $n \mid lm$. Аналогично $m \mid ln$. Однако, поскольку $(m, n) = 1$, получаем, что $n \mid l$ и $m \mid l$, а следовательно, $mn \mid l$. Отсюда и из того, что $l \mid mn$, имеем $l = mn$. Рассмотрим случай, когда $(m, n) > 1$. Разлагая числа m и n на простые множители и упорядочивая последние специальным образом, получаем

$$m = p_1^{e_1} \dots p_s^{e_s} p_{s+1}^{e_{s+1}} \dots p_r^{e_r} \quad (e_i \geqslant 0,\ i = 1, \dots, r),$$

$$n = p_1^{f_1} \dots p_s^{f_s} p_{s+1}^{f_{s+1}} \dots p_r^{f_r} \quad (f_i \geqslant 0,\ i = 1, \dots, r),$$

где

$$e_1 \geqslant f_1, \dots, e_s \geqslant f_s, \quad e_{s+1} < f_{s+1}, \dots, e_r < f_r.$$

Поскольку по предположению $n \nmid m$, то $e_i < f_i$ для некоторого i и, следовательно, $s < r$. Положим $\bar{m} = p_1^{e_1} \dots p_s^{e_s}$ и $\bar{n} = p_{s+1}^{f_{s+1}} \dots p_r^{f_r}$. Поскольку $(\bar{m}, \bar{n}) = 1$, элементы $\bar{a} = a^{m/\bar{m}}$ и $\bar{b} = b^{n/\bar{n}}$ имеют соответственно порядки $\bar{m}$ и $\bar{n}$, а в силу первой части настоящего доказательства порядок элемента $c = \bar{a}\bar{b}$ равен $\bar{m}\bar{n}$. Заметим, что $(m, n) = p_1^{f_1} \dots p_s^{f_s} \cdot p_{s+1}^{e_{s+1}} \dots p_r^{e_r}$. Но тогда $\bar{m}\bar{n} = mn/(m, n)$, т. е. порядок элемента c равен $mn/(m, n)$.

Дополнение. Здесь приводятся некоторые обозначения и краткие сведения из теории чисел, знание которых необходимо для чтения этой и последующих глав. Будем обозначать через $b \mid a$ тот факт, что b делит a; запись $b \nmid a$ означает, что b не делит a. Наибольший общий делитель чисел a и b обозначим через (a, b). Если $(a, b) = 1$, то будем говорить, что числа a и b взаимно просты. Если $n \mid (a - b)$, то говорят, что a сравнимо с b по модулю n, и обозначают это следующим образом: $a \equiv b \pmod{n}$. Таким образом, записи $n \mid (a - b)$ и $a \equiv b \pmod{n}$ эквивалентны. Остаток от деления числа a на целое положительное число n обозначим через $R_n(a)$.

N. 1. Для произвольного целого числа a и произвольного целого положительного числа b существуют такие числа q и r, что

$$a = bq + r, \quad 0 \leqslant r < b,$$

и притом единственные.

N. 2. Если $b \mid a$, то $(a, b) = b$.

N. 3. Если $a = bq + c$, то $(a, b) = (b, c)$.

N. 4. *Алгоритм Евклида.* Этот алгоритм хорошо известен как алгоритм нахождения наибольшего общего делителя.

Предположим, что $a \geqslant b$, и рассмотрим следующую последовательность равенств:

$$\begin{aligned}
a &= bq_1 + r_2, && 0 < r_2 < b,\\
b &= r_2q_2 + r_3, && 0 < r_3 < r_2,\\
r_2 &= r_3q_3 + r_4, && 0 \leqslant r_4 < r_3,\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
r_{n-2} &= r_{n-1}q_{n-1} + r_n, && 0 < r_n < r_{n-1},\\
r_{n-1} &= r_nq_n + r_{n+1}, && 0 = r_{n+1}.
\end{aligned}$$

Число равенств в этой последовательности всегда конечно, поскольку

$$a \geqslant b > r_2 > r_3 > \ldots \geqslant 0.$$

Согласно свойствам N. 2 и N. 3 целых чисел,

$$(a, b) = (b, r_2) = (r_2, r_3) = \ldots = (r_{n-1}, r_n) = r_n.$$

Другими словами, наибольший общий делитель чисел a и b равен последнему ненулевому остатку в вышеприведенной последовательности равенств.

N. 5. Для любых двух целых чисел a и b существуют такие целые числа x и y, что $ax + by = (a, b)$.

Доказательство. Воспользуемся последовательностью равенств, введенной выше при рассмотрении свойства N. 4. А именно подставим r_2 из первого равенства во второе. В полученное таким образом равенство подставим r_3 из третьего равенства. Продолжая эту процедуру, исключим все остатки $r_2, \ldots, r_{n-1}$ и придем к равенству $ax + by = r_n = (a, b)$.

Свойство N. 5 можно также сформулировать в следующем виде:

N. 5′. Если числа a и b взаимно простые, то существуют такие целые числа x и y, что $ax + by = 1$.

N. 6. Если $a_1 \equiv b_1 \pmod m$ и $a_2 \equiv b_2 \pmod m$, то

1) $a_1 + a_2 \equiv b_1 + b_2 \pmod m$,

2) $a_1 - a_2 \equiv b_1 - b_2 \pmod m$,

3) $a_1a_2 \equiv b_1b_2 \pmod m$.

N. 7. Если $(c, m) = 1$ и $ca \equiv cb \pmod m$, то

$$a \equiv b \pmod m.$$

Доказательство. Так как $ca \equiv cb \pmod{m}$, то $m \mid c(a-b)$. Так как m и c взаимно простые, то $m \mid (a-b)$, т. е. $a \equiv b \pmod{m}$.

Функция Эйлера $\varphi(n)$ определяется для всех натуральных n как количество чисел между 1 и n, взаимно простых с n. Положим $\varphi(1) = 1$. Если $n = p^e$, где p — простое число, то среди всех целых чисел от 1 до p^e содержится p^{e-1} чисел, кратных p. Отсюда следует, что

$$\varphi(p^e) = p^e - p^{e-1} = p^e\left(1 - \frac{1}{p}\right).$$

В общем случае, когда $n = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$ ($p_1, \dots, p_r$ — простые числа), имеем [4]

$$\varphi(n) = \varphi\left(p_1^{e_1}\right) \dots \varphi\left(p_r^{e_r}\right) = n\left(1 - \frac{1}{p_1}\right) \dots \left(1 - \frac{1}{p_r}\right).$$

Доказательство последнего равенства здесь опускается [1]).

2.2. Кольца и поля

Группа — это алгебраическая система с одной операцией, а рассматриваемые ниже кольца и поля являются алгебраическими системами с двумя операциями.

Кольца. Пусть R — множество, для любых двух элементов которого определены две операции $+$ и $\cdot$. Алгебраическая система $\langle R, +, \cdot \rangle$ называется *кольцом*, если выполнены следующие три аксиомы:

R. 1. $\langle R, + \rangle$ — коммутативная группа.

R. 2. $\langle R, \cdot \rangle$ — полугруппа, т. е. алгебраическая система, для которой выполняются G. 1 и G. 2.

R. 3. Аксиома дистрибутивности. *Для любых трех элементов* a, b *и* c *из* R

$$a(b + c) = ab + ac,$$
$$(a + b)c = ac + bc.$$

Заметим, что в кольце для операции $\cdot$, вообще говоря, могут не выполняться аксиомы G. 3, G. 4 и AG. 5. Если же операция $\cdot$ коммутативна в кольце, то такое кольцо называется *коммутативным*. Если в кольце существует единичный элемент относительно операции $\cdot$, т. е. если выполняется аксиома G. 3, то это кольцо называется *кольцом с единицей*.

---

[1]) См. гл. 8, разд. 8.1.5. — *Прим. ред.*

Задача 2.15. Доказать следующее утверждение. Алгебраическая система $\langle Z, +, \cdot \rangle$, где $+$ и $\cdot$ есть обычные сложение и умножение целых чисел, является коммутативным кольцом с единицей.

Задача 2.16. Доказать следующее утверждение. Пусть M_n — множество всех квадратных матриц размера $n \times n$, элементами которых являются действительные числа, и пусть $+$ и $\cdot$ есть обычные операции сложения и умножения матриц. Тогда алгебраическая система $\langle M_n, +, \cdot \rangle$ является кольцом с единицей. Это кольцо называется кольцом матриц.

Указание. Единичным элементом относительно операции $\cdot$ в M_n является единичная матрица. Поскольку умножение матриц, вообще говоря, — некоммутативная операция, то кольцо матриц является, таким образом, примером некоммутативного кольца.

Задача 2.17. Пусть дано кольцо R. Множество всех многочленов

$$a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n, \quad a_i \in R, \quad n \geqslant 0,$$

с коэффициентами из R является кольцом относительно обычных операций сложения и умножения многочленов. Это кольцо называется *кольцом многочленов* от неизвестного x над кольцом R и обозначается через $R[x]$.

Подмножество I кольца R называется *идеалом* кольца R, если оно удовлетворяет следующим двум условиям:

I. 1. Для любых двух элементов a и b из I элемент $a + b$ принадлежит I.

I. 2. Для любого элемента x кольца R и любого a из I оба произведения ax и xa лежат в I.

Задача 2.18. Множество целых чисел Z является коммутативным кольцом с единицей. Какой вид имеют идеалы кольца Z?

Решение. Пусть I — некоторый идеал кольца Z, $I \subset Z$, и пусть n — минимальное целое положительное число из I. Для любого элемента $a \in I$ однозначно определяются такие числа q и r, что

$$a = qn + r, \quad 0 \leqslant r < n.$$

Так как $n \in I$, то $(-q)n \in I$. Следовательно, $a + (-q)n = a - qn = r \in I$. Если допустить, что $r > 0$, то сразу получаем противоречие, так как по предположению n — минимальное число в I. Следовательно, $r = 0$ и

$$a = qn.$$

Это означает, что каждый элемент идеала I делится на n. Таким образом, идеал I — это множество всех чисел, кратных n Будем обозначать этот идеал через (n).

В теории групп с помощью нормальных делителей строятся факторгруппы. В теории колец роль нормальных делителей выполняют идеалы.

Если I — идеал кольца R, то в R можно ввести новое бинарное отношение следующим образом:

$$a \equiv b \pmod{I} \Leftrightarrow a - b \in I.$$

Это бинарное отношение является отношением эквивалентности. Для того чтобы убедиться в этом, проверим справедливость аксиом отношения эквивалентности:

1) Так как $a - a = 0 \in I$, то $a \equiv a \pmod{I}$.

2) Пусть $a \equiv b \pmod{I}$. Тогда, поскольку $a - b \in I$ и $(-1) \in R$, имеем $-(a - b) = b - a \in I$, а это означает, что $b \equiv a \pmod{I}$.

3) Пусть $a \equiv b \pmod{I}$ и $b \equiv c \pmod{I}$. Тогда, поскольку $a - b \in I$ и $b - c \in I$, то из I. 1 получаем, что $(a - b) + (b - c) = a - c \in I$. Это означает, что $a \equiv c \pmod{I}$.

Классы эквивалентности, определяемые отношением $\equiv$, называются *классами вычетов* кольца R по модулю I. Операции сложения и умножения классов вычетов кольца R по модулю I определяются следующим образом. Обозначим через (x) и (y) классы вычетов, содержащие соответственно элементы x и y. По определению

$$(x) + (y) = (x + y),$$
$$(x) \cdot (y) = (x \cdot y).$$

Эти определения были бы некорректными, если бы для некоторых элементов x' и y', таких, что $(x) = (x')$ и $(y) = (y')$, оказалось, что $(x + y) \neq (x' + y')$ или $(x \cdot y) \neq (x' \cdot y')$. Однако, как показано ниже, это случиться не может, так что введенные определения корректны. Действительно, пусть x' и y' таковы, что $(x) = (x')$ и $(y) = (y')$. По определению

$$x - x' \in I \quad \text{и} \quad y - y' \in I.$$

Отсюда следует, что

$$(x + y) - (x' + y') = (x - x') + (y - y') \in I,$$
$$xy - x'y' = xy - xy' + xy' - x'y' = x(y - y') + (x - x')y' \in I,$$

а именно

$$(x + y) = (x' + y') \quad \text{и} \quad (xy) = (x'y').$$

Введя таким образом операции $+$ и $\cdot$ в множестве классов вычетов кольца R, порожденных идеалом I, мы построили новое

кольцо. Это кольцо называется *кольцом классов вычетов* и обозначается через R/I.

Утверждение 2.19. *Как следует из задачи 2.18, идеалы кольца Z — это множества вида $(n) = \{nz \mid z \in Z\}$. Выберем одно из чисел n и с помощью идеала $I = (n)$ построим кольцо классов вычетов $Z/(n)$. Аддитивная группа этого кольца изоморфна факторгруппе Z_n из утверждения 2.8. Умножение определяется равенством $(i) \cdot (j) = (i \cdot j) =$ (Остаток от деления $i \cdot j$ на n).*

Задача 2.20. Построить таблицы сложения и умножения для кольца класса вычетов $Z/(6)$.

Решение:

+	(0)	(1)	(2)	(3)	(4)	(5)
(0)	(0)	(1)	(2)	(3)	(4)	(5)
(1)	(1)	(2)	(3)	(4)	(5)	(0)
(2)	(2)	(3)	(4)	(5)	(0)	(1)
(3)	(3)	(4)	(5)	(0)	(1)	(2)
(4)	(4)	(5)	(0)	(1)	(2)	(3)
(5)	(5)	(0)	(1)	(2)	(3)	(4)

·	(0)	(1)	(2)	(3)	(4)	(5)
(0)	(0)	(0)	(0)	(0)	(0)	(0)
(1)	(0)	(1)	(2)	(3)	(4)	(5)
(2)	(0)	(2)	(4)	(0)	(2)	(4)
(3)	(0)	(3)	(0)	(3)	(0)	(3)
(4)	(0)	(4)	(2)	(0)	(4)	(2)
(5)	(0)	(5)	(4)	(3)	(2)	(1)

Как видно из этих таблиц, кольцо $Z/(6)$ относительно операции $\cdot$ не является группой. В частности, в $Z/(6)$ произведение двух ненулевых элементов может оказаться равным (0), например $(2) \cdot (3) = (6) = (0)$. Если произведение двух элементов кольца равно нулю, то каждый из них называется *делителем нуля*. Коммутативное кольцо без делителей нуля, обладающее единичным элементом по умножению, называется *областью целостности*.

Поля. Пусть F — множество, содержащее по крайней мере 2 элемента, такое, что для любых двух элементов a и b из F определены две операции $+$ и $\cdot$. Алгебраическая система $\langle F, +, \cdot \rangle$ называется *полем*, если она удовлетворяет следующим аксиомам:

F. 1. $\langle F, + \rangle$ — коммутативная группа.

Пусть 0 — единичный элемент группы $\langle F, + \rangle$, а F^* — множество, полученное из F удалением элемента 0.

F. 2. $\langle F^*, \cdot \rangle$ — коммутативная группа.

F. 3. Закон дистрибутивности. Для любых трех элементов a, b и c из F

$$a \cdot (b + c) = a \cdot b + a \cdot c,$$

$$(a + b) \cdot c = a \cdot c + b \cdot c.$$

Утверждение 2.21. *Множество R действительных чисел и множество C комплексных чисел являются полями относительно обычных операций сложения и умножения и называются соответственно полем действительных чисел и полем комплексных чисел. Однако коммутативное кольцо с единицей $\langle Z, +, \cdot\rangle$ полем не является.*

Утверждение 2.22. *Свойства полей.* (*Единичный элемент поля по сложению обозначается через* 0.)

1) *Для любого элемента a поля $a \cdot 0 = 0 \cdot a = 0$.*

2) *Для любых двух ненулевых элементов a и b* поля $a \cdot b \neq 0$.

3) *Для любых двух элементов a и b поля* $-(a \cdot b) = (-a) \cdot b = a \cdot (-b)$.

4) *Если $a \cdot b = a \cdot c$ и $a \neq 0$, то $b = c$.*

Доказательство.

1) По определению единичного элемента аддитивной группы $b = b + 0$. В силу закона дистрибутивности $a \cdot b = a \cdot (b+0) = a \cdot b + a \cdot 0$. Так как поле является группой по сложению, то из единственности единичного элемента аддитивной группы (см. задачу 2.4) имеем $a \cdot 0 = 0$. Аналогично доказывается, что $0 \cdot a = 0$.

2) Аксиома F. 2 устанавливает, что множество $F^* = F - \{0\}$ является группой относительно операции $\cdot$. Сформулированное выше свойство 2 поля есть просто следствие свойства замкнутости операции $\cdot$ в группе $\langle F^*, \cdot\rangle$.

3) Справедливость следующих равенств следует непосредственно из определения поля: $0 = 0 \cdot b = (a + (-a)) \cdot b = a \cdot b + (-a) \cdot b$. Из существования и единственности обратного элемента (см. задачу 2.4) имеем $(-a) \cdot b = -(a \cdot b)$. Аналогично доказывается, что $a \cdot (-b) = -(a \cdot b)$.

4) Так как $a \neq 0$, то существует обратный к a элемент по умножению a^{-1}. Умножая обе части равенства $a \cdot b = a \cdot c$ слева на a^{-1}, получаем $b = c$.

Таким образом, поле — это алгебраическая система, в которой можно выполнять, как обычно, сложение, вычитание, умножение и деление элементов.

Алгебраические системы $\langle F, +\rangle$ и $\langle F^*, \cdot\rangle$ поля $\langle F, +, \cdot\rangle$ называются соответственно аддитивной и мультипликативной группами поля. Единичные элементы относительно операций $+$ и $\cdot$ называются соответственно *нулевым элементом* поля *и единичным элементом* поля и обозначаются через 0 и 1 в тех случаях, когда их нельзя спутать с числами 0 и 1. По аналогии с группами число элементов в поле называется порядком поля. Поля, порядок которых конечен, называются *конечными полями*.

Задача 2.23. По определению поле содержит по крайней мере два элемента 0 и 1. Построить таблицы сложения и умножения для поля, состоящего только из двух элементов 0 и 1.

Решение. Сначала воспользуемся свойствами 1 и 2 поля, сформулированными в утверждении 2.19. Три графы таблицы умножения можно заполнить, воспользовавшись первым из этих свойств. В силу второго свойства произведение $1 \cdot 1$ не может быть равно 0, а следовательно, $1 \cdot 1 = 1$. Три графы таблицы сложения, за исключением той, которая соответствует сумме $1 + 1$, можно сразу же заполнить, воспользовавшись тем, что 0 является единичным элементом по сложению. Сумма $1 + 1$ может принимать два значения 0 и 1. Однако, поскольку $\{0, 1\}$ есть группа по сложению, то, согласно свойствам групповых таблиц, $1 + 1 = 0$.

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Примечание. Операция сложения в данном случае представляет собой так называемое сложение по модулю 2. Часто для обозначения этой операции используется знак $\oplus$.

Если порядок поля задан, то степень свободы при составлении таблиц сложения и умножения элементов поля в действительности не столь уж велика, как это могло бы показаться с первого взгляда (см., например, последнюю задачу). Способы построения полей могут быть различными, но если порядок поля фиксирован, то конечное поле определяется однозначно с точностью до изоморфизма. Другими словами, каким бы образом не было построено конечное поле, результат будет одним и тем же. Конечное поле может быть изоморфно другим полям, т. е. может быть представлено различными способами, и выбор удобного представления поля — это в действительности очень важная проблема.

Если существует биективное отображение f поля F на поле F', такое, что для любых двух элементов a и b из F

$$f(a + b) = f(a) + f(b),$$
$$f(a \cdot b) = f(a) \cdot f(b),$$

то поля F и F' называются *изоморфными*. В подобных случаях говорят, что отображение f сохраняет операции $+$ и $\cdot$.

Рассмотрим здесь конкретный способ построения конечных полей.

Утверждение 2.24. *Если p — простое число, то $Z/(p)$ — поле.*

Доказательство. Выше уже было доказано, что $Z/(p)$ — кольцо (утверждение 2.19). Чтобы доказать, что $Z/(p)$ является по-

лем, достаточно показать, что для каждого класса вычетов (a) кольца $Z/(p)$, за исключением класса вычетов (0), существует обратный элемент $(a)^{-1}$. Рассмотрим смежные классы (1), (2), ..., $(p-1)$. Заметим, что если p — простое число, то равенство $(a, p) = 1$ справедливо для всех $a = 1, \ldots, p-1$. Следовательно, в силу свойства N. 5′ целых чисел существуют такие числа x и y, что $ax + py = 1$. Отсюда следует, что $(a) \cdot (x) = (a \cdot x) = (1 - py) = (1)$, т. е. обратным элементом к (a) является элемент (x). Таким образом, если из кольца $Z/(p)$ исключить элемент (0), то оставшиеся элементы образуют мультипликативную группу (очевидно, коммутативную), а это означает, что $Z/(p)$ является полем. Порядок этого поля равен p.

Итак, для каждого простого числа p мы построили конечное поле порядка p. Элементами этого поля являются классы вычетов $(0) = (p)$, (1), ..., $(p-1)$ кольца Z по модулю (p). Операции сложения и умножения элементов этого поля задаются равенствами

$$(i) + (j) = (i + j) = (R_p(i + j)),$$
$$(i) \cdot (j) = (i \cdot j) = (R_p(i \cdot j)),$$

где $R_p(a)$ — остаток от деления числа a на число p.

Задача 2.25. Построить таблицы сложения и умножения элементов конечных полей порядков 3 и 5.

Решение. Для простоты элементы (i) поля будем обозначать просто символом i:

Поле порядка 3

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Поле порядка 5

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Как легко видеть из описанного выше метода построения конкретных полей, для любого простого числа p существует конечное поле порядка p. Если же p не является простым числом,

то построить поле порядка p тем же методом, а именно вводя сложение и умножение классов вычетов кольца Z с помощью операции вычисления остатка при делении чисел на модуль p, нельзя. Это, в частности, связано с тем, что если p — составное число, то произведение двух ненулевых элементов a и b может оказаться равным p. В этом случае $R_p(a \cdot b) = R_p(p) = 0$, а следовательно, в силу свойства 2 полей, сформулированного в утверждении 2.22, множество классов вычетов по модулю p не может быть полем. Однако, как будет показано ниже, для любого простого p и любого целого положительного числа m существует поле порядка p^m. В силу вышеизложенного операции сложения и умножения по модулю p^m для построения полей не используются. Как строить поля порядка p^m, будет описано ниже. Вначале докажем существование полей порядка p^m.

Если подмножество F_0 элементов поля F само является полем, то F_0 называется *подполем* F, а F — *расширением* F_0. Поскольку операции в подполе F_0 те же, что и в поле F, то единичный элемент по сложению 0 и единичный элемент по умножению 1 оба входят в F_0.

Утверждение 2.26. *Необходимыми и достаточными условиями того, чтобы непустое отличное от* $\{0\}$ *подмножество* F_0 *поля* F *было подполем поля* F, *являются следующие:*

1) *если* $a, b \in F_0$, *то* $a - b \in F_0$;
2) *если* $a, b \in F_0$, $b \neq 0$, *то* $ab^{-1} \in F_0$.

Теорема 2.3. *Любое конечное поле является расширением поля* $Z/(p)$, *где* p — *простое число* (*или поля, изоморфного* $Z/(p)$).

Доказательство. Пусть $F_0 = \{0, 1, 1+1, \ldots\}$ — подгруппа (по сложению), порожденная элементом 1 поля F, и пусть m — порядок F_0 (m — порядок элемента 1 в аддитивной группе поля F). Тогда

$$\underbrace{1+1+\ldots+1}_{i \text{ слагаемых}} \neq 0, \quad 1 \leqslant i < m,$$

$$\underbrace{1+1+\ldots+1}_{m \text{ слагаемых}} = 0.$$

Допустим, что m не является простым числом, т. е. $m = m_1 m_2$, где $m_1 > 1$ и $m_2 > 1$ — некоторые положительные числа. Тогда из аксиом дистрибутивности поля получаем

$$\underbrace{(1+1+\ldots+1)}_{m_1}\underbrace{(1+1+\ldots+1)}_{m_2} = \underbrace{1+1+\ldots+1}_{m} = 0,$$

что противоречит свойству 2 полей, сформулированному в утверждении 2.22. Таким образом, m должно быть простым числом. Поставим в соответствие элементу $\underbrace{1+1\ldots+1}_{i}$ поля F

класс вычетов (i) поля $Z/(p)$. Так как это отображение взаимно однозначно и сохраняет операции, то F_0 и $Z/(p)$ изоморфны.

В любом конечном поле F порядок p единичного по умножению элемента 1 как элемента аддитивной группы поля F определяется однозначно и называется *характеристикой поля* F. В поле F характеристики p для любого элемента a поля F сумма $\underbrace{(a+a+\ldots+a)}_{p}$ всегда равна 0, так как

$$\underbrace{a+a+\ldots+a}_{p}=a\cdot(\underbrace{1+1\ldots+1}_{p})=a\cdot 0=0.$$

Более того, для любого ненулевого элемента a поля F минимальное число n, для которого $\underbrace{a+a+\ldots+a}_{n}=0$, равно p. В самом деле, легко проверить, что

$$\underbrace{a+a+\ldots+a}_{n}=0\Leftrightarrow\underbrace{1+1+\ldots+1}_{n}=0.$$

Таким образом, порядок любого элемента аддитивной группы поля F, за исключением элемента 0, равен p.

Задача 2.27. Пусть F — поле характеристики p. Доказать, что для любых двух элементов a и b из F и любого натурального числа m

$$(a+b)^{p^m}=a^{p^m}+b^{p^m},\quad (a-b)^{p^m}=a^{p^m}-b^{p^m}.$$

Решение. Сначала покажем справедливость первого из этих равенств в случае $m=1$. Раскрывая двучлен $(a+b)^p$ по формуле бинома Ньютона, получаем

$$(a+b)^p=a^p+\binom{p}{1}a^{p-1}b+\binom{p}{2}a^{p-2}b^2+\ldots+\binom{p}{1}ab^{p-1}+b^p,$$

где через $\binom{p}{i}a^{p-i}b^i$ обозначена сумма $a^{p-i}b^i+\ldots+a^{p-i}b^i$ $\left(\binom{p}{i}\text{ слагаемых}\right)$, а $\binom{p}{i}$ — биномиальный коэффициент, равный $p!/i!\,(p-i)!$. Так как при $0<i<p$ знаменатель последнего выражения не делится на p, то все биномиальные коэффициенты $\binom{p}{i}$, $0<i<p$, делятся на p. В поле характеристики p для любого элемента a поля сумма $\underbrace{a+a+\ldots+a}_{p}$ при любом натуральном p равна 0, так что все слагаемые в приведенном выше разложении, за исключением двух крайних, равны нулю. Следовательно,

$$(a+b)^p=a^p+b^p.$$

Далее, в случае $m=2$ имеем

$$(a+b)^{p^2}=((a+b)^p)^p=(a^p+b^p)^p=a^{p^2}+b^{p^2}.$$

Аналогично доказывается справедливость первого равенства и при остальных m. Доказать второе равенство читателям предлагается самостоятельно.

Согласно теореме 2.3, любое конечное поле является расширением некоторого поля $Z/(p)$, где p — простое число. В связи с этим возникает вопрос о структуре конечного поля, если последнее рассматривать как расширение некоторого другого поля. Сразу отметим, что расширение F поля F_0 является векторным пространством над полем F_0. Прежде чем доказать это, напомним определение векторного пространства и кратко рассмотрим свойства последнего, а кроме того, введем понятие кольца многочленов. Тем самым мы завершим подготовку к доказательству указанного выше утверждения. Можно, однако, перейти здесь непосредственно к разд. 2.5, возвращаясь к разд. 2.3 и 2.4 в случае необходимости.

2.3. Векторные пространства

Векторные пространства. Определим *векторное пространство* V над полем F как алгебраическую систему, удовлетворяющую указанным ниже аксиомам:

V. 1. V является аддитивной группой.

V. 2. Для любых двух элементов $x\in V$ и $\alpha\in F$ определено произведение αx, являющееся элементом V.

V. 3. Если $x, y\in V$, $\alpha\in F$, то $\alpha(x+y)=\alpha x+\alpha y$.

V. 4. Если $x\in V$, $\alpha, \beta\in F$, то $(\alpha+\beta)x=\alpha x+\beta x$.

V. 5. Если $x\in V$, $\alpha, \beta\in F$, то $(\alpha\beta)x=\alpha(\beta x)$.

V. 6. $1\cdot x=x$ (1 — единичный элемент поля F).

Утверждение 2.28. *Векторное пространство V над полем F обладает следующими свойствами:*

1) $0\cdot x=0$ (*0 в левой части равенства — элемент поля F, а 0 в правой части — элемент V, являющийся единичным элементом аддитивной группы V и называемый нулевым вектором*);

2) $(-1)x=-x$ (*-1 в левой части равенства — элемент поля F, а $-x$ в правой части — элемент V, обратный x*;

3) $\alpha x=0\in V$, $\alpha\neq 0\Leftrightarrow x=0$.

Доказательство. Согласно определению нулевого элемента поля F, для любого элемента $\alpha\in F$ имеет место равенство $0+\alpha=\alpha$. Тогда

$$(0+\alpha)x=0\cdot x+\alpha\cdot x=\alpha\cdot x \text{ (ввиду V. 4).}$$

Так как V является аддитивной группой и в этой группе существует единственный единичный элемент (в данном случае это нулевой вектор 0), то $0 \cdot x = 0$.

Доказательство свойств 2 и 3 опускается.

Утверждение 2.29. *Пусть* $V_n(F)$ *— множество всех последовательностей* $(x_1, \ldots, x_n)$ *длины* n *с компонентами из поля* F, *т. е.*

$$V_n(F) = \{x \mid x = (x_1, \ldots, x_n),\ x_i \in F,\ i = 1, \ldots, n\}.$$

Пусть $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n) \in V_n(F)$, $\alpha \in F$ *и*

$$x + y = (x_1 + y_1, \ldots, x_n + y_n),$$
$$\alpha x = (\alpha x_1, \ldots, \alpha x_n).$$

Тогда $V_n(F)$ *является векторным пространством над полем* F.

Если $e_1 = (1, 0, \ldots, 0)$, $e_2 = (0, 1, \ldots, 0)$, $\ldots$, $e_n = (0, 0, \ldots, 1)$, то

$$x = (x_1, \ldots, x_n) = x_1 e_1 + \ldots + x_n e_n.$$

Рассмотрим векторное пространство V над полем F. Векторы $v_1, v_2, \ldots, v_k$ из V называются *линейно зависимыми*, если в F существуют такие элементы $\alpha_1, \alpha_2, \ldots, \alpha_k$ (по крайней мере один из которых не равен нулю), что

$$\alpha_1 v_1 + \ldots + \alpha_k v_k = 0, \tag{2.5}$$

и *линейно независимыми* в противном случае. А именно векторы $v_1, \ldots, v_k$ являются линейно независимыми, если равенство (2.5) выполняется лишь когда $\alpha_1 = \ldots = \alpha_k = 0$. Максимальное число линейно независимых векторов в V называется *размерностью* векторного пространства V над полем F и обозначается через $\dim V$. Пусть $\dim V = m$ и $v_1, \ldots, v_m$ — линейно независимые векторы из V (по определению в V существует хотя бы одна такая совокупность векторов). Тогда любой вектор v из V можно представить как *линейную комбинацию* векторов из этой совокупности:

$$v = \beta_1 v_1 + \ldots + \beta_m v_m, \tag{2.6}$$

где $\beta_1, \ldots, \beta_m \in F$. Действительно, как следует из определения m, совокупность из $m + 1$ векторов $v, v_1, \ldots, v_m$ линейно зависима, и, следовательно, существуют такие элементы $\gamma_i \in F$, $i = 0, 1, \ldots, m$, что

$$\gamma_0 v + \gamma_1 v_1 + \ldots + \gamma_m v_m = 0$$

и не все γ_i равны нулю (в частности, $\gamma_0 \neq 0$ [1]).

[1]) Поскольку $v_1, \ldots, v_m$ линейно независимы. — *Прим. ред.*

Умножая обе части последнего равенства на элемент $\gamma_0^{-1} \in F$, являющийся обратным по умножению к элементу γ_0, получаем

$$v = -\gamma_0^{-1}\gamma_1 v_1 - \gamma_0^{-1}\gamma_2 v_2 - \ldots - \gamma_0^{-1}\gamma_m v_m.$$

Таким образом, мы показали, что любой вектор v из V можно представить в виде линейной комбинации векторов $v_1, \ldots, v_m$. Любая совокупность таких линейно независимых векторов из V, что любой вектор из V может быть представлен в виде их линейной комбинации, называется *базисом*. Базис векторного пространства V можно выбрать различными способами. Однако можно доказать, что число векторов в любом базисе равно размерности m векторного пространства V. Если размерность векторного пространства V конечна, то подмножество W векторов из V является базисом тогда и только тогда, когда W состоит из $m = \dim V$ линейно независимых векторов.

Утверждение 2.30. *Пусть $v_1, \ldots, v_m$ один из базисов векторного пространства V. Представление (2.6) любого вектора $v \in V$ единственно.*

Доказательство. Предположим, что вектор v из V может быть представлен в виде (2.6) двумя различными способами, а именно $v = a_1 v_1 + \ldots + a_m v_m = b_1 v_1 + \ldots + b_m v_m$. Тогда

$$(a_1 - b_1) v_1 + \ldots + (a_m - b_m) v_m = 0.$$

Так как векторы $v_1, \ldots, v_m$ линейно независимы, то $a_i - b_i = 0$ $(i = 1, \ldots, m)$, т. е. $a_i = b_i$ $(i = 1, \ldots, m)$.

Подмножество W векторного пространства V, удовлетворяющее следующим двум условиям:

1) $w_1, w_2 \in W \Rightarrow w_1 + w_2 \in W$,

2) для любых $\alpha \in F$ и $w \in W$ элемент $\alpha w \in W$,

само является векторным пространством над F и называется *подпространством векторного пространства V.*

Пусть $v_1, \ldots, v_m$ — базис векторного пространства V над полем F, $m = \dim V$. Скалярное произведение $a \cdot b$ двух векторов $a = \sum_{i=1}^{m} a_i v_i$ и $b = \sum_{i=1}^{m} b_i v_i$ определяется равенством

$$a \cdot b = \left(\sum_{i=1}^{m} a_i v_i\right) \cdot \left(\sum_{i=1}^{m} b_i v_i\right) \stackrel{\Delta}{=} \sum_{i=1}^{m} a_i b_i \in F.$$

Если скалярное произведение двух векторов a и b из V равно нулю, то говорят, что эти векторы *ортогональны. Ортогональное дополнение* $W^{\perp}$ подпространства W векторного пространства V определим следующим образом:

$$x \in W^{\perp} \Leftrightarrow x \cdot y = 0 \quad \text{для всех} \quad y \in W,$$

Кроме того, введем сумму $V_1 \oplus V_2$ двух подпространств V_1 и V_2 векторного пространства V:

$$x \in V_1 \oplus V_2 \Leftrightarrow x = y + z, \quad y \in V_1, \quad z \in V_2,$$

Утверждение 2.31. *Пусть W_1 и W_2 — подпространства V. Тогда $W_1 \cap W_2$ также является подпространством V.*

Доказательство. Проверим справедливость свойств 1 и 2 в определении подпространства для $W_1 \cap W_2$:

1) $w_1, w_2 \in W_1 \cap W_2 \Rightarrow w_1, w_2 \in W_1$ и $w_1, w_2 \in W_2 \Rightarrow w_1 + w_2 \in W_1$, $w_1 + w_2 \in W_2 \Rightarrow (w_1 + w_2) \in W_1 \cap W_2$;

2) Свойство 2 проверяется аналогично.

Утверждение 2.32. *Если W_1 и W_2 — подпространства V, то $W_1 \oplus W_2$ также является подпространством V.*

Доказательство. Как и при доказательстве предыдущего утверждения, проверим справедливость свойств 1 и 2 в определении подпространства:

1) $w = w_1 + w_2$, $u = u_1 + u_2 \in W_1 \oplus W_2$, $w_1, u_1 \in W_1$, $w_2, u_2 \in W_2 \Rightarrow w + u = (w_1 + u_1) + (w_2 + u_2) \in W_1 \oplus W_2$;

2) свойство 2 проверяется аналогично.

Утверждение 2.33 *. $\dim (U \cap W) + \dim (U \oplus W) = \dim U + \dim W$.

Доказательство. Пусть $\dim U = k_1$, $\dim W = k_2$ и $\dim (U \cap W) = k_0$. В $U \cap W$ существует базис из k_0 векторов. Присоединив к этим k_0 векторам $k_1 - k_0$ дополнительных векторов, можно построить базис для U. Точно так же, присоединяя к тем же k_0 векторам $k_2 - k_0$ векторов, можно построить базис для W. В качестве базиса для $U \oplus W$ можно взять базис W из k_0 векторов, присоединив к нему $(k_1 - k_0)$ дополнительных векторов из базиса U и $(k_2 - k_0)$ дополнительных векторов из базиса W. Тогда

$$\dim (U \oplus W) = k_0 + (k_1 - k_0) + (k_2 - k_0).$$

Утверждение 2.34. * *Пусть U и W — подпространства векторного пространства V. Тогда $U^\perp \oplus W^\perp = (U \cap W)^\perp$.*

Доказательство. Так как

$$U \cap W \subseteq U, W,$$

$$U^\perp, W^\perp \subseteq (U \cap W)^\perp,$$

то

$$(U^\perp \oplus W^\perp) \subseteq (U \cap W)^\perp.$$

С другой стороны,

$$U^\perp,\ W^\perp \subseteq U^\perp \oplus W^\perp,$$
$$(U^\perp \oplus W^\perp)^\perp \subseteq U^{\perp\perp},\ W^{\perp\perp},$$
$$(U^\perp \oplus W^\perp)^\perp \subseteq U^{\perp\perp} \cap W^{\perp\perp} = U \cap W,$$
$$(U \cap W)^\perp \subseteq ((U^\perp \oplus W^\perp)^\perp)^\perp = U^\perp \oplus W^\perp.$$

Утверждение 2.35. * *Пусть V — векторное пространство размерности n над полем порядка q, U и A — подпространства размерностей j и k соответственно векторного пространства V, $W = U^\perp$ и $B = A^\perp$. Тогда*

$$|W \cap A| = q^{k-j} |U \cap B|,$$

где через $|S|$ обозначено число элементов в множестве S.

Доказательство. $\dim (U^\perp \cap A) = n - \dim (U^\perp \cap A)^\perp$. Согласно предыдущему утверждению,

$$(U^\perp \cap A)^\perp = U^{\perp\perp} \oplus A^\perp = U \oplus A^\perp,$$
$$\dim (U \oplus A^\perp) = \dim U + \dim A^\perp - \dim (U \cap A^\perp).$$

Тогда

$$\dim (W \cap A) = n - \dim U - \dim B + \dim (U \cap B) =$$
$$= k - j + \dim (U \cap B).$$

2.4. Многочлены

Пусть F — поле, а $F[X]$ — множество всех *многочленов*

$$f(X) = a_0 + a_1 X + \ldots + a_k X^k, \quad a_i \in F, \quad i = 0, 1, \ldots, k, \quad (2.7)$$

от переменной X над полем F. Если в (2.7) $a_k \neq 0$, то говорят, что многочлен $f(X)$ имеет степень k. Степень многочлена f будем обозначать через $\deg f$. Множество всех многочленов, имеющих степень k и менее, обозначим через $F^{(k)}[X]$.

Два многочлена $f(X) = \sum_{i=0}^{k} a_i X^i$ и $g(X) = \sum_{i=0}^{l} b_i X^i$ называются равными, если они имеют одну и ту же степень, т. е. $k = l$, а кроме того, $a_i = b_i$ при всех $0 \leqslant i \leqslant k$. (При этом считается, что $X^0 = 1$, где 1 — единичный элемент поля F.) Многочлен, все коэффициенты которого равны нулю, будем называть нулевым многочленом и обозначать через 0.

Задача 2.36. Перечислить все многочлены степени 2 и менее над полем $F = Z/(2) = \{0, 1\}$ и указать их степени.

Решение.

$$\left.\begin{aligned} f_{111}(X) &= 1 + X + X^2 \\ f_{101}(X) &= 1 \quad\quad + X^2 \\ f_{011}(X) &= \quad + X + X^2 \\ f_{001}(X) &= \quad\quad\quad X^2 \end{aligned}\right\} \text{степень 2,}$$

$$\left.\begin{aligned} f_{110}(X) &= 1 + X \\ f_{010}(X) &= \quad\quad X \end{aligned}\right\} \text{степень 1,}$$

$$f_{100}(X) = 1 \quad\quad - \text{ степень 0,}$$

$$f_{000}(X) = 0 - \text{степень не определена.}$$

Примечание. Степень нулевого многочлена не определена, но когда говорят о многочленах степени, меньшей некоторого целого k, то обычно к ним относят также и нулевой многочлен.

Если в многочлен $f_{011} = X + X^2$ из последней задачи подставлять вместо X элементы 0 и 1 поля F, то он будет принимать те же значения, что и многочлен $f_{000}(X) = 0$. Однако как многочлены $f_{000}(X)$ и $f_{011}(X)$ различны. Заметим, что число различных многочленов степени k и меньше над полем порядка q равно q^{k+1}.

Введем далее операции сложения и умножения многочленов следующим образом. Для любых двух многочленов $f(X) = \sum f_i X^i$ и $g(X) = \sum g_i X^i$ положим

$$f(X) + g(X) = \sum (f_i + g_i) X^i, \tag{2.8}$$

$$f(X) \cdot g(X) = \sum_i \left(\sum_{j=0}^{i} f_j g_{i-j} \right) X^i. \tag{2.9}$$

В частности, если $\alpha \in F$, то

$$\alpha f(X) = \sum_i (\alpha f_i) X^i. \tag{2.10}$$

Введя таким образом сложение и умножение в $F[X]$, мы получили, как нетрудно проверить, кольцо, называемое кольцом многочленов.

Утверждение 2.37. *Многочлены из* $F[X]$ *обладают следующими свойствами:*

1) $f(X) \cdot 0 = 0$ *для любого* $f(X) \in F[X]$;
2) *если* $f(X) \neq 0$ *и* $g(X) \neq 0$, *то* $f(X)g(X) \neq 0$;
3) $-[f(X)g(X)] = [-f(X)]g(X) = f(X)[-g(X)]$;
4) $[f(X) + g(X)]h(X) = f(X)h(X) + g(X)h(X)$;
5) *если* $f(X) \neq 0$ *и* $f(X)g(X) = f(X)h(X)$, *то* $g(X) = h(X)$.

Доказательство. Свойства 1—4 можно доказать точно так же, как и соответствующие свойства из утверждения 2.28. Докажем свойство 5. Из предположений и свойства 4 непосредственно следует, что $0 = f(X)g(X) - f(X)h(X) = f(X)[g(X) - h(X)]$. Так как $f(X) \neq 0$, то в силу свойства 2 $g(X) - h(X) = 0$, так что $h(X) = g(X)$.

Примечание. Метод доказательства свойства 4 из утверждения 2.22 здесь неприменим, так как элемент, обратный к $f(X)$, может и не существовать.

Утверждение 2.38. *Для любых двух ненулевых многочленов* $f(X), g(X) \in F[X]$, *таких, что* $f(X) + g(X) \neq 0$,

$$\deg(f + g) \leqslant \max(\deg f, \deg g),$$

$$\deg(f \cdot g) = \deg f + \deg g.$$

Утверждение 2.39. *Для любых двух многочленов* $f(X)$, $g(X) \in F[X]$, $g(X) \neq 0$, *существуют, и притом единственные, многочлены* $a(X), r(X) \in F[X]$, *такие, что*

$$f(X) = a(X)g(X) + r(X), \quad \deg g > \deg r. \tag{2.1}$$

Доказательство. Для нахождения частного $a(X)$ и остатка $r(X)$ следует провести деление $f(X)$ на $g(X)$ точно так же, как производится деление многочленов над полем действительных чисел. Докажем единственность разложения (2.11). Предположим, что существуют два частных $a(X)$ и $a'(X)$ и два остатка $r(X)$ и $r'(X)$, такие, что

$$a(X)g(X) + r(X) = a'(X)g(X) + r'(X).$$

Тогда

$$g(X)[a(X) - a'(X)] = r'(X) - r(X).$$

Поскольку степень многочлена в правой части последнего равенства меньше, чем степень $g(X)$, то это равенство может иметь место лишь в том случае, когда $a(X) - a'(X) = 0$, и, следовательно, $r(X) - r'(X) = 0$.

Если в формуле (2.11) остаток $r(X)$ равен нулю, то говорят, что $g(X)$ делит $f(X)$. Если в $F[X]$ нет ни одного многочлена степени, большей 0, который делил бы $f(X)$ (за исключением скалярных кратных $f(X)$), то $f(X)$ называется *неприводимым*. Элемент α поля F, такой, что $f(\alpha) = 0$, называется *корнем* $f(X)$. В этом случае также говорят, что уравнение $f(X) = 0$ имеет корень α в поле F.

Утверждение 2.40. *Если уравнение* $f(X) = 0$, $f(X) \in F^k[X]$, *имеет корень* α *в поле* F, *то существует такой многочлен* $g(X) \in F^{k-1}[X]$, *что* $f(X) = (X - \alpha)g(X)$. *Если степень* $f(X)$ *равна* k, *то уравнение* $f(X) = 0$ *имеет не более* k *корней.*

Доказательство. Из предыдущего утверждения следует существование и единственность таких многочленов $g(X) \in F^{k-1}[X]$ и $r(X) \in F$ (поскольку $\deg r < 1$), что

$$f(X) = (X - \alpha)\, g(X) + r(X), \quad r(X) \in F.$$

Подставляя в обе части последнего равенства $X = \alpha$, получаем

$$f(\alpha) = 0 \cdot g(\alpha) + r(\alpha) = r(\alpha).$$

Поскольку α является корнем $f(X)$, то $f(\alpha) = r(\alpha) = 0$, т. е. $r(X) = 0$. Следовательно,

$$f(X) = (X - \alpha)\, g(X).$$

Далее, если уравнение $f(X) = 0$ имеет ν различных корней $\alpha_1, \ldots, \alpha_\nu \in F$, то существует такой многочлен $h(X) \in F^{k-\nu}[X]$, что $f(X) = (X - \alpha_1) \ldots (X - \alpha_\nu) h(X)$. Это означает, что $f(X)$ имеет не более k корней.

Многочлены $f(X)$, $g(X) \in F[X]$ называются *взаимно простыми*, если в $F[X]$ нет ни одного многочлена степени, большей чем 1, который делил бы одновременно и $f(X)$ и $g(X)$.

Утверждение 2.41. *Для двух произвольных взаимно простых многочленов $f(X)$, $g(X) \in F[X]$ существуют такие многочлены $a(X)$ и $b(X)$, что*

$$a(X) f(X) + b(X) g(X) = 1. \tag{2.12}$$

Это свойство многочленов аналогично свойству N. 5′ целых чисел. Доказательство последнего было основано на алгоритме Евклида для целых чисел, однако для доказательства утверждения 2.41 мы воспользуемся другим методом. Этим методом можно было бы доказать и свойство N. 5′ целых чисел точно так же, как и для доказательства вышеприведенного утверждения можно было бы воспользоваться алгоритмом Евклида для многочленов. Для доказательства существования обратных элементов в полях по умножению нам будет вполне достаточно свойства многочленов, сформулированного в утверждении 2.41. Однако в тех случаях, когда обратный элемент для того или иного конкретного элемента требуется найти в явном виде, необходимо использовать непосредственно алгоритм Евклида.

Доказательство. Пусть $I = \{a(X) f(X) + b(X) g(X) \mid a(X), b(X) \in F[X]\} - \{0\}$ и $h(X)$ — один из многочленов минимальной степени из I. Предположим, что $\deg h(X) \geqslant 1$ и в I имеется хотя бы один многочлен, который не делится на $h(X)$. Тогда остаток от деления этого многочлена на $h(X)$ принадлежит I и имеет степень, меньшую степени $h(X)$, что противоречит тому, что $h(X)$ — многочлен минимальной степени в I. Следовательно, все многочлены из I должны делиться на $h(X)$. Это означает, в

частности, что многочлены $f(X) \in I$ и $g(X) \in I$ оба делятся на многочлен $h(X)$ ненулевой степени. Мы получили противоречие, поскольку по условию $h(X)$ и $g(X)$ взаимно просты. Таким образом, любой многочлен минимальной степени в I является многочленом нулевой степени, т. е. элементом поля F. Пусть c — один из таких элементов, $c \in F$. По определению

$$c = a_0(X) f(X) + b_0(X) g(X),$$

где $a_0(X)$, $b_0(X) \in F[X]$. Умножая обе части последнего равенства на c^{-1}, получим равенство (2.12).

Многочлены из $F[X]$, коэффициенты при наивысшей степени X которых равны 1, называются *нормированными многочленами.*

Утверждение 2.42. *Любой многочлен степени 1 и выше над полем F можно представить в виде произведения элемента поля F и нормированных неприводимых многочленов над тем же полем. Это разложение единственно с точностью до порядка следования сомножителей* (*доказательство см. в* [3]).

Выше были рассмотрены основные свойства многочленов из $F[X]$. Поскольку $F[X]$ является кольцом, можно рассматривать кольца классов вычетов, порождаемые идеалами кольца $F[X]$. По существу именно так строятся конечные поля, однако вопросы построения конечных полей будут рассмотрены в следующем разделе. Для построения конечных полей необходимы неприводимые многочлены и следующая задача посвящена нахождению некоторых из них.

Задача 2.43. Найти все неприводимые многочлены степени 4 и менее над полем $Z/(2)$.

Решение. Многочлены первой степени. Имеется два многочлена первой степени X и $X + 1$. По определению оба они неприводимы.

Многочлены второй степени. Многочлен $p(X) = X^2 + aX + b$ второй степени неприводим тогда и только тогда, когда он не делится ни на X, ни на $X - 1$ [1]). Это эквивалентно тому, что $p(0) = b \neq 0$ и $p(1) = 1 + a + b \neq 0$. Следовательно, лишь один из многочленов второй степени, а именно $X^2 + X + 1$, является неприводимым.

Многочлены третьей степени. Многочлен $p(X) = X^3 + aX^2 + bX + c$ является неприводимым тогда и только тогда, когда он не делится ни на X, ни на $X + 1$. Это эквивалентно тому, что $p(0) = c \neq 0$ и $p(1) = 1 + a + b + c \neq 0$, или

[1]) Заметим, что $X - 1 = X + 1$. — *Прим. ред.*

$c = 1$ и $a + b = 1$. Следовательно, имеются два неприводимых многочлена третьей степени $X^3 + X^2 + 1$ и $X^3 + X + 1$.

Многочлены четвертой степени. Многочлен $p(X) = X^4 + aX^3 + bX^2 + cX + d$ является неприводимым тогда и только тогда, когда он не делится ни на один из неприводимых многочленов первой или второй степени. Поскольку имеется только один неприводимый многочлен второй степени, то все многочлены четвертой степени, не являющиеся неприводимыми, за исключением многочлена $(X^2 + X + 1)^2 = X^4 + X^2 + 1$, должны делиться на один из многочленов первой степени. Следовательно, для неприводимых многочленов четвертой степени $p(0) = d \neq 0$, $p(1) = 1 + a + b + c + d \neq 0$. Многочленами, удовлетворяющими этим условиям, являются следующие: $X^4 + X^2 + 1$, $X^4 + X^3 + X^2 + 1$, $X^4 + X + 1$ и $X^4 + X^3 + 1$. Все они, за исключением $X^4 + X^2 + 1$, являются неприводимыми.

Примечание. Таблица неприводимых многочленов до 34-й степени включительно над полем порядка 2 приведена в [1]. В [2] описан способ нахождения неприводимых многочленов и подсчета числа последних. В приложении в конце данной книги дана таблица неприводимых многочленов до 14-й степени включительно над полем порядка 2, до 6-й степени над полем порядка 3 и до 4-й степени над полем порядка 5.

Задача 2.44. Показать, что многочлен X^2+1, рассматриваемый как многочлен над полем $Z/(3)$ порядка 3, является неприводимым.

Решение. В данном случае необходимо проверить, что X^2+1 не делится ни на один из многочленов первой степени. Для этого достаточно убедиться в том, что $f((0)) \neq (0)$, $f((1)) \neq (0)$ и $f((2)) \neq (0)$. Действительно,

$$f((0)) = (0) \cdot (0) + (1) = (1),$$
$$f((1)) = (1) \cdot (1) + (1) = (2),$$
$$f((2)) = (2) \cdot (2) + (1) = (2).$$

Производная многочлена. Как мы уже видели выше, для построения полей $Z/(p)$ используются классы вычетов по простому модулю. Такой прием, когда некоторое множество элементов разбивается на классы по некоторому модулю, используется в алгебре чрезвычайно эффективно. Путем разбиения кольца многочленов на классы вычетов по модулю некоторого неприводимого многочлена в следующем разделе строятся поля порядка p^m (p — простое число). В заключение этого раздела введем понятие производной многочлена и рассмотрим некоторые ее свой-

ства. Приведенные здесь результаты используются при доказательстве различных утверждений следующего раздела.

Заменим в многочлене $f(X)$ переменную X на $X+Y$. Если теперь рассматривать $f(X+Y)$ как многочлен от Y, то коэффициентами последнего будут многочлены от X:

$$f(X+Y)=\sum f_i(X)Y^i.$$

Если считать эквивалентными многочлены, сравнимые по модулю Y^2, то

$$f(X+Y)\equiv f_0(X)+f_1(X)Y \pmod{Y^2},$$

где $f_0(X)=f(X+0)=f(X)$. Обозначив $f_1(X)$ через $f'(X)$, получим

$$f(X+Y)\equiv f(X)+f'(X)Y \pmod{Y^2}.$$

Введенный таким образом многочлен $f'(X)$ называется производной $f(X)$.

Утверждение 2.45. *Для любого элемента $a\in F$ и любых многочленов f, $g\in F[X]$ имеют место следующие равенства:*

1) $(af)'=af'$, $(a)'=0$, $(X)'=1$,
2) $(f+g)'=f'+g'$,
3) $(f\cdot g)'=f'\cdot g+f\cdot g'$,
4) *если* $f(X)=\sum a_iX^i$, *то* $f'(X)=\sum ia_iX^{i-1}$.

Доказательство. Справедливость первых двух равенств проверяется легко. Докажем третье равенство:

$$f(X+Y)\equiv f(X)+f'(X)\cdot Y \pmod{Y^2},$$
$$g(X+Y)\equiv g(X)+g'(X)\cdot Y \pmod{Y^2},$$
$$f(X+Y)\cdot g(X+Y)\equiv (f(X)+f'(X)Y)(g(X)+$$
$$+g'(X)Y) \pmod{Y^2}\equiv f(X)g(X)+(f'(X)g(X)+$$
$$+f(X)g'(X))Y \pmod{Y^2}.$$

Третье равенство доказано. В частности, если $f(X)=X^n$ $(n\geqslant 2)$, то из третьего равенства легко получить, что

$$(X+Y)^n\equiv X^n+nX^{n-1}Y \pmod{Y^2}.$$

Отсюда в свою очередь следует, что

5) $(X^i)'=iX^{i-1}$ $(i\geqslant 1)$.

Из первого, второго и пятого равенств получаем четвертое равенство.

Задача 2.46. Пусть F — поле характеристики 2. Найти производную многочлена

$$f(X)=\sum a_iX^i\in F[X].$$

Решение. Так как характеристика поля коэффициентов равна двум, то из четвертого равенства имеем

$$f'(X)=\sum ia_iX^{i-1}=\sum_{\text{по четным } i} a_iX^{i-1}.$$

Например, если $f(X)=1+X+X^3+X^7+X^8$, то

$$f'(X)=1+X^2+X^6$$

и $\deg f'=6<(\deg f)-1=7$. Этот пример показывает, что правила вычисления производной многочлена с коэффициентами из конечного поля несколько отличаются от обычных правил дифференцирования в поле действительных чисел. Однако равенства 1—5 сохраняются без изменений. Таким образом, понятие производной многочлена можно ввести, не прибегая к предельному переходу, а основываясь лишь на понятии сравнимости многочленов. Этот метод часто называют формальным дифференцированием.

Утверждение 2.47. *Пусть F — поле, а $g(X)\in F[X]$ — неприводимый многочлен, такой, что $g'(X)\neq 0$. Многочлен $f(X)\in F[X]$ делится на $g(X)^2$ тогда и только тогда, когда наибольший общий делитель $(f(x), f'(X))$ многочленов $f(X)$ и $f'(X)$ делится на $g(X)$.*

Доказательство. Допустим, что $f(X)=g(X)h(X)$. Тогда

$$f'(X)=g'(X)h(X)+g(X)h'(X).$$

Если $g(X)\,|\,f'(X)$, то $g(X)\,|\,g'(X)h(X)$. Однако, поскольку $g(X)\nmid g'(X)$ (так как $\deg g>\deg g'$), то $g(X)\,|\,h(X)$, т. е. $g(X)^2\,|\,f(X)$. Это означает, что необходимым и достаточным условием того, что $(g(X))^2$ делит $f(X)$, является наличие общего множителя $g(X)$ в разложениях $f(X)$ и $f'(X)$, что эквивалентно сформулированному выше утверждению.

Утверждение 2.48. *Пусть F — конечное поле порядка q. Каков бы ни был двучлен $X^{q^l}-X\in F[X]$, в $F[X]$ не существует ни одного неприводимого многочлена $g(X)$, такого, что $(g(X))^2\,|\,(X^{q^l}-X)$.*

Доказательство. Так как $(X^{q^l}-X)'=-1$, то справедливость утверждения 2.48 непосредственно следует из утверждения 2.47. Доказанное утверждение означает, что двучлен $X^{q^l}-X$ не имеет кратных корней.

Утверждение 2.49*. *Многочлен*

$$f(X_1,\ldots,X_n)=a_1X_1^{\alpha_{11}}X_2^{\alpha_{12}}\ldots X_n^{\alpha_{1n}}+$$

$$+a_2X_1^{\alpha_{21}}X_{a_{22}}\ldots X_n^{\alpha_{2n}}+\ldots+a_kX_1^{\alpha_{k1}}X_2^{\alpha_{k2}}\ldots X_n^{\alpha_{kn}},\quad 0\leqslant\alpha_{ij}<q,$$

над полем $F = GF(q)$ равен 0 только в том случае, если все его коэффициенты a_i равны 0 [1]).

Доказательство. Пусть R — произвольное кольцо. Тогда ясно, что многочлен $g(X) = a_0 + a_1X + \ldots + a_mX^m \in R[X]$ может быть равен 0 только в том случае, если $a_0 = a_1 = \ldots = a_m = 0$. Докажем сформулированное выше утверждение индукцией по числу неизвестных. Группируя слагаемые $f(X_1, \ldots, X_n)$, имеющие одинаковые степени X_n, запишем равенство $f(X_1, \ldots, X_n) = 0$ в виде

$$A_1(X_1, \ldots, X_{n-1}) X_n^{m_1} + A_2(X_1, \ldots, X_{n-1}) X_n^{m_2} + \ldots$$
$$\ldots + A_l(X_1, \ldots, X_{n-1}) X_n^{m_l} = 0.$$

Каждый коэффициент многочлена от X_n в левой части последнего равенства является элементом кольца $F[X_1, \ldots, X_{n-1}]$. Если это кольцо рассматривать как кольцо R, а X_n — как X, то из сформулированного в начале доказательства утверждения будет следовать, что

$$A_1(X_1, \ldots, X_{n-1}) = A_2(X_1, \ldots, X_{n-1}) = \ldots$$
$$\ldots = A_l(X_1, \ldots, X_{n-1}) = 0.$$

Но тогда по предположению индукции все коэффициенты многочленов $A_i(X_1, \ldots, X_{n-1})$ должны быть равны 0, а следовательно, равны 0 и коэффициенты a_i.

2.5. Конечные поля

Выше мы несколько отклонились от изучения полей, перейдя к рассмотрению векторных пространств и колец многочленов. Однако сейчас, когда все необходимые понятия уже введены, попытаемся более детально рассмотреть структуру конечных полей.

Теорема 2.4. *Конечное поле F является векторным пространством над любым своим подполем F_0.*

Доказательство. Возьмем любой элемент $x_1 \in F - \{0\}$ и рассмотрим множество $F_1 = \{a_1x_1 \mid a_1 \in F_0\}$, $F_1 \subset F$. Если $F - F_1 \neq \phi$, то выберем произвольным образом элемент $x_2 \in F - F_1$ и рассмотрим множество $F_2 = \{a_1x_1 + a_2x_2 \mid a_1, a_2 \in F_0\}$. Если продолжить эту процедуру дальше, то в силу конечности поля F при некотором целом m множество $F_m = \{a_1x_1 + \ldots + a_mx_m \mid a_i \in F_0,\ 1 \leqslant i \leqslant m\}$ совпадает с F. По построению элементы

[1]) Здесь предполагается, конечно, что приведены все подобные члены, т. е. наборы степеней $(\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{in})$ различны при различных i. — *Прим. ред.*

$x_1, \ldots, x_m$ поля F линейно независимы. Следовательно, поле F является векторным пространством размерности m над полем F_0, в качестве базиса которого можно взять элементы $x_1, \ldots, x_m$.

Выбрав в качестве F_0 поле $Z/(p)$ из утверждения 2.24, легко доказать следующую теорему.

Теорема 2.5. *Если F — поле характеристики p, то его порядок является степенью p.*

Доказательство. Как было показано при доказательстве предыдущей теоремы, поле F всегда можно представить в виде $F = \{a_1x_1 + \ldots + a_mx_m \mid a_i \in F_0,\ 1 \leqslant i \leqslant m\}$. Поскольку каждый элемент x поля F можно представить в виде линейной комбинации базисных элементов единственным образом (см. утверждение 2.30), то число различных векторов в F, очевидно, равно p^m, где p — порядок F_0, в данном случае совпадающий с характеристикой поля. Отсюда следует доказательство теоремы.

Как мы уже знаем, характеристика любого конечного поля должна быть простым числом. Выше было доказано, что порядок конечного поля характеристики p равен p^m, где m — некоторое целое число ($m \geqslant 1$).

Из теоремы 2.4, в частности, следует, что любое конечное поле является векторным пространством над некоторым полем $Z/(p)$, порядок которого есть простое число. Элементы этого векторного пространства, называемые ниже векторами, можно представить различными способами. Например, их можно представить в виде последовательностей из n элементов поля (утверждение 2.29). С другой стороны, каждому вектору $(a_0, a_1, \ldots, a_{n-1})$ можно сопоставить многочлен

$$a_0 + a_1X + a_2X^2 + \ldots + a_{n-1}X^{n-1}.$$

Если ввести операции сложения многочленов и умножения многочлена на константу посредством соотношений (2.8) и (2.10), то эти операции будут в точности соответствовать сложению и умножению на константу соответствующих последовательностей символов длины n. Это позволяет рассматривать множество многочленов степени $n-1$ и менее и последовательности символов длины n как различные представления одного и того же векторного пространства. Представление произвольной бесконечной последовательности $(s_0, s_1, \ldots)$ элементов некоторого множества S в виде формального многочлена

$$s_0 + s_1X + s_2X^2 + \ldots$$

является хорошо известным и широко применяемым приемом. Такие многочлены иногда называют *производящими функциями* или *формальными степенными рядами.* При представлении по-

следовательностей в виде формальных многочленов символ X используется лишь для указания положения элементов в исходной последовательности и не является некоторым «неизвестным элементом». В силу этого символ X был назван формальной переменной. Как мы уже знаем, конечное поле F является векторным пространством размерности k над полем F_0, порядок которого есть простое число. Поэтому любой элемент (вектор) поля F можно представить в виде многочлена степени $k-1$ над полем F_0. Однако само по себе это представление будет не очень интересным, если не указать, каким образом следует складывать и умножать элементы поля F, представленные в виде многочленов. Выше мы ввели сложение векторов, но не касались их умножения. Если представить векторы в виде многочленов, то можно ввести операцию умножения векторов, используя для этого умножение многочленов. Однако, как показывает доказательство свойства 5 из утверждения 2.37, при таком определении умножения элементы $F[X]$ могут не иметь обратных, т. е. $F[X]$ не является полем. Более того, при представлении элементов поля F в виде многочленов степени $k-1$ из $F_0^{k-1}[X]$ при обычном умножении многочленов из $F_0^{k-1}[X]$ может получиться многочлен степени k или выше, а это означает, что обычное умножение многочленов оказывается не замкнутой в $F_0^{k-1}[X]$ операцией. Ключом к построению конечных полей порядка p^m ($m \geqslant 1$, p — простое число) является разложение кольца многочленов на классы вычетов по некоторому специально выбранному модулю. Аналогичный прием использовался и при построении полей вида $Z/(p)$, когда были введены операции по модулю p.

Построение полей путем разложения кольца многочленов на классы вычетов по модулю неприводимого многочлена

Пусть $g(X) \in F[X]$ — некоторый неприводимый многочлен степени m. Используя $g(X)$, разложим $F[X]$ на классы следующим образом. Отнесем к одному классу все многочлены, которые дают один и тот же остаток при делении на $g(X)$. Класс, содержащий многочлен $f(X)$, обозначим через $(f(X))$. Два многочлена $f_1(X)$ и $f_2(X)$ принадлежат одному классу тогда и только тогда, когда $f_1(X)-f_2(X)$ делится на $g(X)$. Это мы будем записывать в виде $f_1(X) \equiv f_2(X) \pmod{g(X)}$. Полученные таким образом классы будем называть классами вычетов по модулю $g(X)$. Множество всех классов вычетов по модулю $g(X)$ обозначим через $F[g]$. Поскольку существует взаимно однозначное соответствие между классами вычетов из $F[g]$ и многочленами из $F^{m-1}[X]$, то ясно, что число классов вычетов равно $|F|^m$. Введем в $F[g]$ операции сложения и умножения

следующим образом. Для любых $(f(X))$, $(h(X)) \in F[g]$ положим

$$(f(X)) + (h(X)) = (f(X) + h(X)),$$
$$(f(X)) \cdot (h(X)) = (f(X) \cdot h(X)).$$

При таком определении сложения и умножения классы вычетов (1) и (0) оказываются соответственно единичным и нулевым элементами $F[g]$. Пусть $(f(X)) \neq 0$. Так как многочлены $f(X)$ и $g(X)$ взаимно просты, то, согласно утверждению 2.41, существует единственный многочлен $a(X)$, такой, что

$$a(X) f(X) = 1 - b(X) g(X) \equiv 1 \pmod{g(X)}.$$

Следовательно,

$$(a(X)) \cdot (f(X)) = (a(X) \cdot f(X)) = (1).$$

Это означает, что в $F[g]$ существует элемент $(a(X))$, обратный к $(f(X))$. Следовательно, $F[g]$ — поле. В частности, если $F = Z/(p)$, p — простое число, то, допустив существование в $F[X]$ неприводимого многочлена степени m, мы доказали существование поля порядка p^m (существование неприводимых многочленов для любых m и p будет следовать из приведенной ниже теоремы 2.11).

Описанный выше способ построения полей порядка p^m в общем аналогичен рассмотренному ранее способу построения полей вида $Z/(p)$. Действительно, множество $(g(X))$ многочленов в $F[X]$, которые делятся на неприводимый многочлен $g(X)$, является одним из идеалов кольца $F[X]$. Кольцо классов вычетов $F[X]/(g(X))$, порожденное идеалом $(g(X))$ кольца $F[X]$, как было показано выше, есть поле. Неприводимый многочлен выполняет в данном случае ту же роль, что и простое число p в кольце целых чисел.

Обозначим через α класс вычетов (X), содержащий многочлен X. Пусть $g(X) = \sum_{i=0}^{m} g_i X^i$. Тогда

$$\begin{aligned} g_0 + g_1\alpha + \ldots + g_m\alpha^m &= g_0 + g_1(X) + \ldots + g_m(X^m) = \\ &= (g_0 + g_1 X + \ldots + g_m X^m) = \\ &= (g(X)) = (0) = \text{Нулевой элемент.} \end{aligned}$$

Это означает, что многочлен $g(X)$ имеет корень α в поле $F[g]$. Многочлен $g(X)$ является неприводимым над полем F, и уравнение $g(X) = 0$ не имеет решений в поле F. Однако, как мы только что показали, уравнение $g(X) = 0$ имеет решение в поле $F[g]$. В этом смысле поле $F[g]$ является расширением поля F,

полученным присоединением к F корня уравнения $g(X)=0$. Каждый элемент β поля $F[g]$ можно представить в виде

$$\beta = a_{m-1}\alpha^{m-1} + \ldots + a_1\alpha + a_0, \quad a_i \in F, \quad 1 \leqslant i \leqslant m.$$

Это так называемое *многочленное представление* элементов поля. Кроме того, выбрав в качестве базиса элементы α^{m-1}, α^{m-2}, ..., 1 поля $F[g]$, каждый элемент β поля $F[g]$ можно представить также в виде последовательности

$$(a_{m-1}, a_{m-2}, \ldots, a_0).$$

Последнее представление называется *векторным представлением* элементов поля.

Задача 2.50. Показать, что элементы α^{m-1}, α^{m-2}, ..., 1 образуют базис $F[g]$.

Построим несколько конечных полей указанным выше способом.

Задача 2.51. Построить поле порядка 3^2.

Решение. В качестве неприводимого многочлена второй степени над $F = Z/(3)$ возьмем многочлен X^2+1 (задача 2.44). Каждый элемент поля $F[g]$ имеет вид $(a+bX)$, где $a, b \in F$. Найдем сумму и произведение элементов $(a+bX)$, $(c+dX) \in F[g]$:

$$(a+bX)+(c+dX) = (a+c+(b+d)X),$$
$$(a+bX)\cdot(c+dX) = (ac+(bc+ad)X+bdX^2) =$$
$$= (ac-bd)+(bc+ad)X.$$

Эти формулы напоминают соответствующие формулы для вычисления суммы и произведения комплексных чисел. Отличие состоит лишь в том, что в последних вместо X обычно стоит мнимая единица i.

С точки зрения алгебры поле комплексных чисел полностью идентично описанному выше полю, поскольку оно может быть построено путем разбиения на классы вычетов по модулю неприводимого над полем действительных чисел многочлена X^2+1 кольца многочленов над полем действительных чисел. Поэтому формулы для нахождения суммы и произведения элементов этих полей полностью идентичны.

В поле комплексных чисел используется соотношение $i^2=-1$, т. е. $i^2+1=0$. Понижение степеней слагаемых вида i^a, $a>1$, с помощью этого соотношения эквивалентно вычислению остатка при делении на i^2+1. При этом в поле комплексных чисел для обозначения корня уравнения $X^2+1=0$ используется специальный символ i.

Задача 2.52. Построить поле порядка 2^3, используя неприводимый многочлен $X^3 + X + 1$.

Решение. Положим $g(X) = X^3 + X + 1$ и рассмотрим следующие восемь классов вычетов по модулю $g(X)$:

$$\{(0), (1), (X), (X+1), (X^2), (X^2+1), (X^2+X), (X^2+X+1)\}.$$

Определим сложение и умножение этих классов вычетов следующим образом:

$$(f(X)) + (h(X)) = (f(X) + h(X)),$$

$$(f(X)) \cdot (h(X)) = \text{Остаток от деления } f(X)h(X) \text{ на } g(X).$$

Поскольку складывать элементы легко, ниже приведена лишь таблица умножения элементов введенного поля. В таблице элементы поля представлены в виде многочленов. Элемент 0 из этой таблицы исключен.

$\cdot$	1	α	$\alpha+1$	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$
1	1	α	$\alpha+1$	α^2	α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$
α	α	α^2	$\alpha^2+\alpha$	$\alpha+1$	1	$\alpha^2+\alpha+1$	α^2+1
$\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$	α^2+1	$\alpha^2+\alpha+1$	α^2	1	α
α^2	α^2	$\alpha+1$	$\alpha^2+\alpha+1$	$\alpha^2+\alpha$	α	α^2+1	1
α^2+1	α^2+1	1	α^2	α	$\alpha^2+\alpha+1$	$\alpha+1$	$\alpha^2+\alpha$
α^2+1	$\alpha^2+\alpha$	$\alpha^2+\alpha+1$	1	α^2+1	$\alpha+1$	α	α^2
$\alpha^2+\alpha+1$	$\alpha^2+\alpha+1$	α^2+1	α	1	$\alpha^2+\alpha$	α^2	$\alpha+1$

Составление таблиц, подобных приведенной выше, является довольно трудоемким. Представление элементов поля в виде многочленов или векторов удобно с точки зрения простоты сложения и, наоборот, неудобно с точки зрения умножения. Избежать затруднений, связанных с умножением элементов поля, представленных в виде многочленов или векторов, позволяет *представление элементов в виде степеней* некоторого элемента этого поля. Если с помощью вышеприведенной таблицы найти многочленные представления элементов поля 1, α, α^2, ..., то можно построить другую таблицу:

Степенное представление	Многочленное представление
$1 =$	1
$\alpha =$	α
$\alpha^2 = \alpha^2$	
$\alpha^3 =$	$\alpha + 1$
$\alpha^4 = \alpha^2 +$	α
$\alpha^5 = \alpha^2 +$	$\alpha + 1$
$\alpha^6 = \alpha^2$	$+ 1$
$(\alpha^7 =$	$1)$

Итак, все ненулевые элементы поля оказались представленными в виде степеней элемента α. В этом случае произведение любых двух элементов поля, представленных соответственно в виде α^i и α^j, можно найти следующим простым способом:

$$\alpha^i \cdot \alpha^j = \alpha^{i+j} = \alpha^{R_7(i+j)},$$

где $R_7(i+j)$ — остаток от деления $i+j$ на 7 (здесь мы пользуемся свойством ассоциативности мультипликативной группы поля и тем, что $\alpha^7 = 1$). Как мы только что видели, в случае представления элементов поля в виде степеней одного из них умножение осуществляется очень просто. При наличии таблицы соответствия различных представлений можно обойтись без таблиц сложения и умножения. Нулевой элемент поля иногда по традиции обозначают через α^∞. Описанный выше способ умножения элементов поля применим только в том случае, когда все ненулевые элементы поля могут быть представлены в виде степеней одного из них. Однако, как следует из приведенной ниже теоремы, такое представление элементов в виде степеней всегда возможно.

Теорема 2.6. *В любом конечном поле F найдется по крайней мере один элемент α, такой, что все элементы $F^* = F - \{0\}$ могут быть представлены в виде степеней α^i $(1 \leqslant i \leqslant |F|)$ элемента α. Другими словами, F^* — циклическая группа. Элемент α, степени которого пробегают все ненулевые элементы поля, называется примитивным.*

Доказательство. Пусть α — произвольный элемент мультипликативной группы F^* поля F порядка m (по определению m все элементы $\alpha, \alpha^2, \ldots, \alpha^m$ различны и $\alpha^m = 1$). Если $m = |F^*|$, то теорема доказана. Предположим, что $m < |F^*|$. Тогда, так как элементы F^*, порядок которых делит m, являются корнями уравнения $X^m - 1 = 0$, то их число не превышает m (утверждение 2.40). Следовательно, в F^* существует элемент, порядок которого не является делителем m. Пусть β — один из таких элементов порядка n. Согласно утверждению 2.14, в F^* существует элемент порядка $mn/(m, n)$. Заметим, что $mn/(m, n) > m$. Повторяя эту процедуру достаточное количество раз, всегда можно доказать существование в F^* элемента порядка $|F^*|$.

Теорема 2.7. *Любой элемент конечного поля порядка p^m является корнем уравнения $X^{p^m} - X = 0$.*

Доказательство. Пусть α — один из примитивных элементов поля F, а β — произвольный элемент $F^* = F - \{0\}$. Элемент β можно представить в виде некоторой степени элемента α, т. е.

$\beta = \alpha^i$. Тогда $\beta^{p^m-1} = \alpha^{i(p^m-1)} = 1$. Следовательно, любой элемент β из F^* является корнем уравнения $X^{p^m-1} - 1 = 0$.

Утверждение 2.53. *Пусть a — целое положительное число, n — минимальное целое положительное число, такое, что $a \mid (q^n - 1)$. Тогда*

1) $a \mid (q^m - 1) \Leftrightarrow n \mid m$;

2) $(q^n - 1) \mid (q^m - 1) \Leftrightarrow n \mid m$;

3) $(a, q) = 1 \Leftrightarrow$ *существует такое n, что* $a \mid (q^n - 1)$.

Доказательство. Пусть $m = ln + r$, $0 \leqslant r < n$. Тогда $q^m - 1 = (q^{ln} - 1)q^r + q^r - 1$. Так как $(q^n - 1) \mid (q^{ln} - 1)$, то $a \mid (q^{ln} - 1)$. Если $r \neq 0$, то $a \nmid (q^m - 1)$, поскольку по определению $a \nmid (q^r - 1)$. Второе утверждение следует из первого.

Чтобы доказать третье утверждение, рассмотрим следующую последовательность целых чисел: $R_a(1)$, $R_a(q)$, $R_a(q^2)$, ..., где $R_a(i)$ — остаток от деления i на a. Так как число различных остатков при делении на a не может превышать a, то найдутся такие i и j $(i < j)$, что $R_a(q^i) = R_a(q^j)$, т. е. $q^i \equiv q^j \pmod a$. Так как $(a, q) = 1$, то в силу свойства N.7 целых чисел $1 \equiv q^{j-i} \pmod a$, т. е. $a \mid (q^{j-i} - 1)$. Таким образом, мы доказали, что если $(a, q) = 1$, то существует такое целое n, что $a \mid (q^n - 1)$. Использованный здесь метод доказательства будет применяться в дальнейшем неоднократно. Доказательство обратного утверждения предоставляется провести читателю самостоятельно.

Запись $g(X) \mid h(X)$, часто используемая ниже, означает, что многочлен $g(X)$ делит многочлен $h(X)$.

Утверждение 2.54.

1) $(X^n - 1) \mid (X^m - 1) \Leftrightarrow n \mid m$.

Пусть $g(X) \in F[X]$ и n — минимальное целое число, такое, что $g(X) \mid (X^n - 1)$ (n называется показателем $g(X)$). Тогда

2) $g(X) \mid (X^m - 1) \Leftrightarrow n \mid m$.

Пусть F — поле, F_0 — подполе поля F и α — элемент мультипликативной группы $F - \{0\}$ поля F. Обозначим через I_α множество всех многочленов над F_0, корнем которых является α, а именно

$$I_\alpha = \{f(X) \in F_0[X] - \{0\} \mid f(\alpha) = 0\}.$$

Если порядок F равен p^m, то, согласно теореме 2.7 $\left(X^{p^{m-1}} - X\right) \in I\alpha$. Пусть $m_\alpha(X)$ — нормированный многочлен минимальной степени в I_α; этот многочлен называется *минимальным многочленом* элемента α над F_0. Минимальный многочлен $m_\alpha(X)$

является неприводимым. Действительно, если предположить, что $m_\alpha(X) = f_1(X)f_2(X)$, $f_1(X), f_2(X) \in F_0(X)$, $\deg f_1 \geqslant 1$ и $\deg f_2 \geqslant 1$. то из определения $m_\alpha(X)$ непосредственно будет следовать, что $m_\alpha(\alpha) = f_1(\alpha)f_2(\alpha) = 0$. Но тогда в силу свойства 2 поля из утверждения 2.22 либо $f_1(\alpha) = 0$, либо $f_2(\alpha) = 0$, т. е. в I_α содержится многочлен, степень которого меньше, чем степень $m_\alpha(X)$, т. е. мы получили противоречие. Далее заметим, что все многочлены из I_α должны делиться на $m_\alpha(X)$. Действительно, допустим, что в I_α содержится многочлен $f(X)$, который не делится на $m_\alpha(X)$. Тогда, как мы знаем, существуют многочлены $q(X)$ и $r(X)$, такие, что

$$f(X) = m_\alpha(X)q(X) + r(X), \quad \deg r < \deg m_\alpha.$$

Подставим в последнее равенство вместо X элемент α. Тогда, так как $r(\alpha) = 0$, опять получаем, что в I_α содержится многочлен, степень которого меньше, чем степень $m_\alpha(X)$, что противоречит определению $m_\alpha(X)$. Следовательно, все многочлены из I_α — это многочлены вида $am_\alpha(X)$, $a \in F_0[X] - \{0\}$. В частности, отсюда следует, что минимальный многочлен элемента α определяется однозначно.

Теорема 2.8. *Пусть F_0 — поле порядка q, $g(X) \in F_0[X]$ — нормированный неприводимый многочлен степени m и n — показатель $g(X)$. Тогда m — минимальное целое число, такое, что $n \mid (q^m - 1)$. В частности, если $n = q^m - 1$, то $g(X)$ называется примитивным многочленом.*

Доказательство. Пусть $F = F_0[g]$ и α — элемент (X) поля F. Как мы уже видели выше, $g(\alpha) = 0$. Так как $g(X)$ — неприводимый многочлен, то $m_\alpha(X) = g(X)$. Согласно теореме 2.7, α является корнем уравнения $X^{q^m-1} - 1 = 0$, так что $g(X) \mid (X^{q^m-1} - 1)$. Отсюда и из утверждения 2.54 следует, что $n \mid (q^m - 1)$. Допустим, что $n \mid (q^l - 1)$, $0 < l < m$. Тогда $\alpha^{q^l} = \alpha$, поскольку $g(X) \mid (X^{q^l-1} - 1)$. Как указывалось выше, любой элемент β поля $F_0[g]$ может быть представлен в виде

$$\beta = a_0 + a_1\alpha + \ldots + a_{m-1}\alpha^{m-1}, \quad a_i \in F_0 \quad (0 \leqslant i < m).$$

Из теоремы 2.7 имеем $a_i^q = a_i$. Поскольку, кроме того, $\alpha^{q^l} = \alpha$, получаем (см. задачу 2.27)

$$\beta^{q^l} = a_0^{q^l} + a_1^{q^l}\alpha^{q^l} + \ldots + a_{m-1}^{q^l}\alpha^{(m-1)q^l} =$$
$$= a_0 + a_1\alpha + \ldots + a_{m-1}\alpha^{m-1} = \beta.$$

Это означает, что порядок любого ненулевого элемента поля $F_0[g]$ является делителем $q^l - 1$. Получили противоречие. Следовательно, $l = m$.

Теорема 2.9. *Пусть F_0 — поле порядка q и пусть $g(X)$ — неприводимый многочлен степени m из $F_0[X]$. Тогда*

$$g(X) \mid (X^{q^l-1} - 1) \Leftrightarrow m \mid l.$$

Доказательство. Согласно утверждению 2.53,

$$m \mid l \Leftrightarrow (q^m - 1) \mid (q^l - 1).$$

Но тогда из утверждения 2.54 получаем

$$m \mid l \Leftrightarrow (X^{q^m-1} - 1) \mid (X^{q^l-1} - 1).$$

Из теоремы 2.8 следует, что минимальное k, для которого $g(X) \mid (X^{q^k-1} - 1)$, равно m.

Из единственности разложения любого многочлена на неприводимые множители (утверждение 2.42), из того, что многочлен $X^{q^m} - X$ не имеет кратных множителей (утверждение 2.48), и из теоремы 2.9 следует справедливость следующей теоремы.

Теорема 2.10. *Пусть F_0 — поле порядка q. Тогда двучлен $X^{q^m} - X$ равен произведению различных неприводимых нормированных многочленов из $F_0[X]$, степени которых делят m.*

Обозначим через N_i число неприводимых нормированных многочленов степени i, входящих в $F_0[X]$. Очевидно, что $N_1 = q$. Из теоремы 2.10 непосредственно получаем, что для любого целого $l \geqslant 2$

$$q^l = \sum_{i \mid l} iN_i,$$

где сумма берется по всем делителям i числа l (включая 1 и l). Отсюда следует, что

$$q^l \geqslant iN_i \left(\text{поскольку } q^l = \sum_{j \mid l} jN_j = iN_i + \sum_{\substack{j \mid l \\ j \neq i}} jN_j \geqslant iN_i \right).$$

Но тогда

$$q^l = lN_l + \sum_{\substack{i \mid l \\ i \neq l}} iN_i \leqslant lN_l + \sum_{i=1}^{l/2} iN_i \leqslant lN_l + q + \ldots$$

$$\ldots + q^{l/2} < lN_l + lq^{l/2},$$

откуда

$$N_l > (q^l - lq^{l/2})/l \tag{2.13}$$

и

$$N_l > 0 \quad (l \geqslant 1).$$

Таким образом, для любого целого положительного числа m в $F_0[X]$ существует неприводимый многочлен степени m. От-

сюда и описанного выше способа построения полей следует справедливость следующей теоремы.

Теорема 2.11. *Для любого простого числа p и любого целого положительного m существует конечное поле порядка p^m.*

В случае конечных групп, как мы уже видели выше (задача 2.4), из равенства порядков групп еще не следует их изоморфизм. Однако все поля одного и того же порядка изоморфны. Докажем это.

Пусть F — произвольное конечное поле, являющееся расширением поля F_0. Пусть $|F_0| = q = p^m$ и $|F| = q^l$. Возьмем произвольный многочлен $g(X)$ из $F_0[X]$, степень которого является делителем l (как следует из теорем 2.4 и 2.10, это всегда можно сделать). Этот многочлен $g(X)$ является минимальным многочленом $m_\alpha(X)$ некоторого элемента α поля F. Пусть n — степень $g(X)$. Рассмотрим следующее подмножество $F_0(\alpha)$ поля F:

$$F_0(\alpha) = \{a_1 + a_2\alpha + \ldots + a_n\alpha^{n-1} \mid a_i \in F_0,\ 1 \leqslant i \leqslant n\}.$$

Сопоставим каждому элементу $a_1 + a_2\alpha + \ldots + a_n\alpha^{n-1} \in F_0(\alpha)$ элемент $(a_1 + a_2X + \ldots + a_nX^{n-1}) \in F_0[g]$. Заметим, что это отображение φ является взаимно однозначным, и если

$$A = a_1 + a_2\alpha + \ldots + a_n\alpha^{n-1} \in F_0(\alpha),$$
$$B = b_1 + b_2\alpha + \ldots + b_n\alpha^{n-1} \in F_0(\alpha),$$

то

$$A + B = (a_1 + b_1) + (a_2 + b_2)\alpha + \ldots + (a_n + b_n)\alpha^{n-1} \in F_0(\alpha),$$

т. е. отображение φ сохраняет операцию сложения. Пусть $g(X) = g_0 + g_1X + \ldots + X^n$. Поскольку $g(\alpha) = 0$, то

$$\alpha^n = -g_0 - g_1\alpha - \ldots - g_{n-1}\alpha^{n-1} \in F_0(\alpha).$$

Покажем, что все элементы вида α^i, $i \geqslant n$, поля F принадлежат $F_0(\alpha)$. Так,

$$\alpha^{n+1} = \alpha \cdot \alpha^n = -g_0\alpha - g_1\alpha^2 - \ldots - g_{n-1}\alpha^n =$$
$$= g_0g_{n-1} + (g_1g_{n-1} - g_0)\alpha + \ldots + (g_{n-1}g_{n-1} - g_{n-2})\alpha^{n-1} \in F_0(\alpha).$$

Аналогично можно доказать, что и остальные элементы α^i, $i > n$, принадлежат $F_0(\alpha)$. Отсюда следует, что если A, $B \in F_0(\alpha)$, то $A \cdot B \in F_0(\alpha)$. С другой стороны, в $F_0[g]$

$$(X^n) = (g_0 + g_1X + \ldots + g_{n-1}X^{n-1}),$$

и, следовательно, отображение φ сохраняет также и операцию умножения. Таким образом, $F_0(\alpha)$ и $F_0[g]$ изоморфны. Элементы $\varphi(0)$ и $\varphi(1)$ являются соответственно нулевым и единичным элементами $F_0[g]$. Поэтому для любого $A \in F_0(\alpha)$ элементы

$-A$ и A^{-1} (если $A \neq 0$) также принадлежат $F_0(\alpha)$. Следовательно, $F_0(\alpha)$ является подполем поля F_0. Порядок $F_0(\alpha)$ равен q^n и n является одним из делителей l. Поле $F_0(\alpha)$ является минимальным подполем поля F, содержащим F_0 и α. В частности, если α выбрать таким образом, что $n = l$, то $F_0(\alpha) = F$. Полагая $F_0 = Z/(p)$, где p — простое число, получаем следующую теорему.

Теорема 2.12. *Все конечные поля одного и того же порядка изоморфны. Другими словами, конечное поле заданного порядка единственно с точностью до обозначения элементов.*

Поле порядка p^m будем обозначать через $GF(p^m)$ (первые две буквы являются сокращением английского термина Galois Field).

Следовательно, $GF(p^m)$ изоморфно $F_0[g]$ (здесь $F_0 = Z/(p)$, p — простое число, а $g(X)$ — нормированный неприводимый многочлен степени m над $F_0[X]$). Этот изоморфизм имеет место при любом выборе многочлена $g(X)$. Однако различным многочленам $g(X)$ соответствуют различные представления. Выше было доказано, что если поле существует, то оно единственно. Далее рассмотрим некоторые свойства полей, которые потребуются нам в дальнейшем.

Теорема 2.13. *Пусть F — поле, являющееся расширением поля F_0, $|F_0| = q$, $|F| = q^m$, и пусть $\beta \in F - \{0\}$. Тогда показатель минимального многочлена $m_\beta(X)$ элемента β равен порядку ν элемента β. Степень n многочлена $m_\beta(X)$ является минимальным целым положительным числом, таким, что $\nu \mid (q^n - 1)$; при этом $n \mid m$. Корнями уравнения $m_\beta(X) = 0$ являются следующие n различных элементов: β, β^q, ..., $\beta^{q^{n-1}}$.*

Доказательство. Так как $\beta^\nu - 1 = 0$, то $m_\beta(X) \mid (X^\nu - 1)$. С другой стороны, если $m_\beta(X) \mid (X^l - 1)$, то $\beta^l = 1$. Следовательно, показатель $m_\beta(X)$ равен ν. Из теоремы 2.8 следует, что степень многочлена $m_\beta(X)$ является минимальным целым положительным числом, таким, что $\nu \mid (q^n - 1)$. Кроме того, из теоремы 2.6 следует, что $\nu \mid (q^m - 1)$. Но тогда, согласно утверждению 2.53, $n \mid m$. Пусть $m_\beta(X) = \sum_{i=0}^{n} c_i X^i$. Тогда, так как $m_\beta(\beta) = \sum_{i=0}^{n} c_i \beta^i$ и

$$(m_\beta(\beta))^q = \sum_{i=0}^{n} c_i^q (\beta^i)^q = \sum_{i=0}^{n} c_i (\beta^q)^i = m_\beta(\beta^q) = 0,$$

то β^q также является корнем уравнения $m_\beta(X) = 0$. Аналогично можно доказать, что все элементы β, β^q, β^{q^2}, ... являются корнями уравнения $m_\beta(X) = 0$. Среди этих корней элементы

$\beta, \beta^q, \ldots, \beta^{q^{n-1}}$ являются различными. Для того чтобы убедиться в справедливости последнего утверждения, предположим противное, а именно что $\beta^{q^i} = \beta^{q^j}$, $0 \leqslant i < j < n$. В этом случае

$$\beta = \beta^{q^m} = \left(\beta^{q^j}\right)^{q^{m-j}} = \left(\beta^{q^i}\right)^{q^{m-j}} = \beta^{q^{m+i-j}},$$

и, следовательно, порядок ν элемента β делит $q^{m+i-j} - 1$. Но тогда, согласно утверждению 2.53, $n \mid (m + i - j)$. С другой стороны, так как $n \mid m$ и $m - n < m - j + i \leqslant m - 1$, то $n \nmid (m - j + i)$, и мы пришли к противоречию.

Как следует из теоремы 2.13 и доказательства теоремы 2.8, неприводимый многочлен $g(X)$ степени m из $F_0[X]$ имеет m различных корней в поле $F_0[g]$, т. е. в силу теоремы 2.12 в поле $GF(q^m)$. Таким образом, многочлен $g(X)$ не имеет корней в поле F_0 и в то же время имеет m различных корней в $F_0[g]$.

Утверждение 2.55. *Пусть $m_\alpha(X)$ — минимальный многочлен элемента α. Если β является корнем уравнения $m_\alpha(X) = 0$, то $m_\beta(X) = m_\alpha(X)$. Все корни минимального многочлена $m_\alpha(X)$ имеют один и тот же порядок.*

Доказательство. Если $m_\alpha(\beta) = 0$, то $m_\beta(X) \mid m_\alpha(X)$. Тогда, так как $m_\alpha(X)$ — неприводимый многочлен, то $m_\beta(X) = m_\alpha(X)$. Если многочлен $m_\alpha(X)$ имеет степень n, то его корнями являются элементы $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}}$. Пусть $|F| = q$ и $m_\alpha(X) \in F[X]$. Тогда, если элемент α имеет порядок ν, а элемент α^{q^j} — порядок ν', то

$$\left(\beta^{q^j}\right)^{\nu} = \beta^{\nu q^j} = \left(\beta^{\nu}\right)^{q^j} = 1^{q^j} = 1, \text{ так что } \nu \mid \nu',$$

$$\beta^{\nu'} = \left(\beta^{q^n}\right)^{\nu'} = \beta^{q^i q^{n-i} \nu'} = \left(\left(\beta^{q^j}\right)^{\nu'}\right)^{q^{n-j}} = 1^{q^{n-j}} = 1,$$

так что $\nu' \mid \nu$. Следовательно, $\nu = \nu'$.

Утверждение 2.56. *Пусть α — произвольный элемент из $F^* = F - \{0\}$. Тогда, если порядок α равен ν, то порядок μ элемента α^i равен $\nu/(i, \nu)$.*

Доказательство следует из утверждения 2.2 и теоремы 2.6.

Утверждение 2.57. *Пусть $|F| = q$. Тогда F имеет ровно $\varphi(q-1)$ примитивных элементов (φ — функция Эйлера).*

Доказательство. Множество $F^* = F - \{0\}$ является циклической группой порядка $q - 1$. Порядок любого элемента из F^* является делителем $q - 1$. Порядок μ элемента a^i равен $(q-1)/(i, q-1)$. Отсюда видно, что $\mu = q - 1$ тогда и только

тогда, когда $(i, q-1)=1$, но количество целых чисел i, для которых выполняется последнее неравенство, равно по определению $\varphi(q-1)$.

Пусть $f(X)=\sum_{i=0}^{m} c_i X^i$ — многочлен степени m. Многочлен $f^*(X)=\sum_{i=0}^{m} c_{m-i} X^i$ называется *двойственным* к многочлену $f(X)$. Многочлен $f(X)$, для которого $f^*(X)=f(X)$, называется *самодвойственным*. Если $f(X)=\Pi(X-\alpha_i)$, то $f^*(X)=\Pi(X-\alpha_i^{-1})$, где α_i^{-1} — элемент, обратный к α_i.

Утверждение 2.58.

1) *Многочлен $f(X)$ неприводим тогда и только тогда, когда неприводим многочлен $f^*(X)$.*

2) *Многочлен $f(X)$ является примитивным тогда и только тогда, когда примитивным является многочлен $f^*(X)$.*

Пусть l и q — взаимно простые целые положительные числа. Пусть $0 \leqslant i < l$ и целые числа i, iq, ..., iq^{n-1} попарно несравнимы по модулю l (т. е. среди остатков от деления этих чисел на l нет одинаковых), но $iq^j \equiv iq^n \pmod{l}$ при некотором $0 \leqslant j \leqslant n-1$. Так как $(q, l)=1$, то $iq^{n-j} \equiv i \pmod{l}$ (здесь мы воспользовались свойством N.7 целых чисел). Следовательно, при сделанных выше предположениях $j=0$. Если числа i, iq, ..., iq^{n-1} заменить на их остатки от деления на l, то полученная таким образом совокупность чисел называется (q, l)-*циклом*, содержащим i. Количество чисел, входящих в цикл, называется *длиной цикла*. Длина n цикла, содержащего i, представляет собой минимальное целое положительное число, удовлетворяющее уравнению $iq^n \equiv i \pmod{l}$.

Пусть $F=GF(q)$. Разобьем числа от 0 до q^m-1 на (q, q^m-1)-циклы. Пусть i_1, i_2, ..., i_n — циклы, содержащие i. Возьмем некоторый примитивный элемент α поля $GF(q^m)$ и рассмотрим минимальный многочлен $m_\beta(X) \in F[X]$ элемента $\beta=\alpha^i$. Тогда, согласно теореме 2.13, корни уравнения $m_\beta(X)=0$ — это элементы α^{i_1}, α^{i_2}, ..., α^{i_n}. Степень $m_\beta(X)$ равна длине цикла. Таким образом, мы установили соответствие между циклами и минимальными многочленами (фиксировав предварительно примитивный элемент).

Задача 2.59. Рассмотрим поле $GF(2^4)$. Используя неприводимый над $GF(2)$ многочлен четвертой степени $g(X)=X^4+X+1$, составить таблицу соответствия представлений элементов поля $GF(2^4)$ в виде степеней примитивного элемента и многочленов. Найти минимальный многочлен элемента $\alpha=(X)$.

Решение. Таблицу соответствия следует составлять последовательно так, как это было описано при доказательстве теоремы 2.12:

$$\begin{array}{llll}
\alpha^0 = & & 1 & \alpha^8 = \alpha^2 + 1 \\
\alpha^1 = & \alpha & & \alpha^9 = \alpha^3 + \alpha \\
\alpha^2 = & \alpha^2 & & \alpha^{10} = \alpha^2 + \alpha + 1 \\
\alpha^3 = & \alpha^3 & & \alpha^{11} = \alpha^3 + \alpha^2 + \alpha \\
\alpha^4 = & \alpha + 1 & & \alpha^{12} = \alpha^3 + \alpha^2 + \alpha + 1 \\
\alpha^5 = & \alpha^2 + \alpha & & \alpha^{13} = \alpha^3 + \alpha^2 + 1 \\
\alpha^6 = & \alpha^3 + \alpha^2 & & \alpha^{14} = \alpha^3 + 1 \\
\alpha^7 = & \alpha^3 + \alpha + 1 & & (\alpha^{15} = 1)
\end{array}$$

Как видно из доказательства теоремы 2.8, минимальным многочленом $m_\alpha(X)$ элемента α является сам многочлен $g(X) = X^4 + X + 1$. Однако $m_\alpha(X)$ можно найти также следующим образом. Так как, согласно теореме 2.13, все корни $m_\alpha(X)$ — это элементы α, α^2, α^4, α^8, то

$$\begin{aligned}
m_\alpha(X) &= (X - \alpha)(X - \alpha^2)(X - \alpha^4)(X - \alpha^8) = \\
&= X^4 - (\alpha + \alpha^2 + \alpha^4 + \alpha^8) X^3 + (\alpha^3 + \alpha^5 + \alpha^9 + \alpha^6 + \\
&\quad + \alpha^{10} + \alpha^{12}) X^2 - (\alpha^7 + \alpha^{11} + \alpha^{13} + \alpha^{14}) X + \alpha^{15} = \\
&= X^4 - 0 \cdot X^3 + 0 \cdot X^2 - X + 1 = X^4 + X + 1.
\end{aligned}$$

Задача 2.60. В случае $q = 2$ и $m = 4$, разбив совокупность целых чисел от 0 до $2^4 - 2$ на $(2, 2^4 - 1)$-циклы и выбрав в качестве минимального многочлена, соответствующего циклу, содержащему 1, многочлен $q(X) = X^4 + X + 1$, найти минимальные многочлены, соответствующие другим циклам.

Решение. Совокупность целых чисел от 0 до $2^4 - 2$ разбивается на циклы следующим образом:

$$\begin{array}{llll}
0 & & & \\
1, & 2, & 4, & 8 \\
3, & 6, & 12, & 9 \\
5, & 10 & & \\
7, & 14 & 13, & 11
\end{array}$$

Циклу, содержащему 0, соответствует минимальный многочлен $X + 1$. Циклу, содержащему 3, соответствует минимальный мно-

гочлен

$$(X-\alpha^3)(X-\alpha^6)(X-\alpha^{12})(X-\alpha^9)=$$
$$=X^4-(\alpha^3+\alpha^6+\alpha^9+\alpha^{12})X^3+(\alpha^9+\alpha^{12}+\alpha^{15}+$$
$$+\alpha^{15}+\alpha^3+\alpha^6)X^2-(\alpha^6+\alpha^3+\alpha^9+\alpha^{12})X+1=$$
$$=X^4+X^3+X^2+X+1 \quad (\text{так как } \alpha^{15}=1)$$

(поскольку в этой и предыдущей задачах поле $GF(2^4)$ строится с помощью одного и того же многочлена, то для вычисления суммы $\alpha^3+\alpha^6+\alpha^9+\alpha^{12}$ в данном случае можно воспользоваться таблицей соответствия из предыдущей задачи).

Полученные выше многочлены являются самодвойственными. Поэтому наряду с корнем α^i они имеют в качестве корня также и α^{-i}. Так как в данном случае $\alpha^3\cdot\alpha^{12}=\alpha^6\cdot\alpha^9=\alpha^{15}=1$, то $\alpha^{-3}=\alpha^{12}$ и $\alpha^{-6}=\alpha^9$. Из таблицы циклов легко видеть, что минимальный многочлен, соответствующий циклу, содержащему числа 5 и 10, также является самодвойственным и имеет степень 2. Имеются два многочлена, удовлетворяющие этому условию: X^2+1 и X^2+X+1. Однако $X^2+1=(X+1)^2$, т. е. многочлен X^2+1 не является неприводимым. Следовательно, минимальным многочленом, соответствующим циклу, содержащему 5, является X^2+X+1. Наконец, рассмотрим последний цикл, содержащий число 14. Поскольку $\alpha^{14}=\alpha^{-1}$, то минимальные многочлены, соответствующие циклу, содержащему 1, и циклу, содержащему 14, двойственны друг другу. Следовательно, минимальным многочленом, соответствующим циклу, содержащему 14, является многочлен X^4+X^3+1.

В приведенном выше примере

$$X^{16}+X=X(X+1)(X^2+X+1)(X^4+X+1)(X^4+X^3+$$
$$+X^2+X+1)(X^4+X^3+1).$$

Это соответствует случаю, когда в теореме 2.12 $q=2$ и $m=4$. Порядок каждого сомножителя является делителем числа 4.

Найдем порядки элементов α^i, когда числа i принадлежат различным циклам. Из утверждения 2.55 получаем

0 : 1
1, 2, 4, 8: $15=15/(2, 15)=15/(4, 15)=15/(8, 15)$
3, 6, 12, 9: $5=15/(3, 15)=15/(6, 15)=15/(12, 15)=15/(9, 15)$
5, 10 : $3=15/(5, 15)=15/(10, 15)$
7, 14, 13, 11: $15=15/(7, 15)=15/(14, 15)=15/(13, 15)=15/(11, 15)$

Пусть $\alpha^0, \alpha^1, \ldots, \alpha^{n-1}$ — совокупность корней уравнения X^n-1. Разбив эту совокупность на группы элементов, имеющих

один и тот же порядок, и введя многочлены

$$\psi_d(X) = \prod_{i,\ \text{порядок}\ \alpha^i = d} (X - \alpha^i),$$

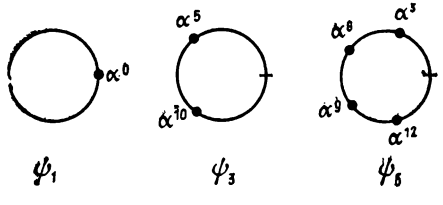
получим

$$X^n - 1 = \prod_{d \mid n} \psi_d(X). \tag{2.14}$$

Многочлены, имеющие своими корнями все элементы одного и того же порядка, называются *круговыми*. В приведенном выше примере

$$X - 1 = \psi_1(X),$$
$$X^3 - 1 = \psi_1(X)\psi_3(X), \quad \psi_3(X) = X^2 + X + 1,$$
$$X^5 - 1 = \psi_1(X)\psi_5(X), \quad \psi_5(X) = X^4 + X^3 + X^2 + X + 1,$$
$$X^{15} - 1 = \psi_1(X)\psi_3(X)\psi_5(X)\psi_{15}(X),$$
$$\psi_{15}(X) = (X^4 + X + 1)(X^4 + X^3 + 1).$$

В общем случае круговые многочлены представляют собой произведения минимальных многочленов. Согласно теореме 2.13,



Фиг. 2.1.

ψ_d является минимальным многочленом в том и только том случае, если степень ψ_d является минимальным целым числом n, таким, что $d \mid (q^n - 1)$. В приведенном выше примере ψ_1, ψ_3, ψ_5 — минимальные многочлены. Однако поскольку минимальное число n, такое, что $15 \mid (2^n - 1)$ равно 4, то $\psi_{15}(X)$ представляет собой произведение двух неприводимых многочленов.

Разделим окружность на 15 равных частей с помощью 15 точек, которые обозначим в порядке их расположения на окружности через α^0, α^1, ..., α^{14}. Исследуем взаимосвязь между положением на окружности корней многочленов ψ_1, ψ_3, ψ_5 и ψ_{15} и полученными выше соотношениями (фиг. 2.1).

Составляя таблицу циклов, как показано на фиг. 2.1, любой двучлен $X^n - 1$ можно разложить в произведение неприводимых многочленов. Рассмотрим пример.

Задача 2.61. Разложить на множители многочлен $X^8 - 1$ над полем $GF(3)$.

Решение. Сначала построим все $(3, 8)$-циклы:

$$\begin{array}{l} 0 \\ 1\ \ 3 \\ 2\ \ 6 \\ 4 \\ 5\ \ 7 \end{array}$$

Двучлен $X^8 - 1$, рассматриваемый как многочлен над полем $GF(3^2)$, имеет своими корнями все ненулевые элементы этого поля. Поле $GF(3^2)$ было построено в задаче 2.51. Однако, поскольку ни один из корней неприводимого многочлена $X^2 + 1$, использованного в этой задаче, не является примитивным элементом поля $GF(3^2)$, то сначала нужно найти один из таких элементов. Возьмем элемент $\beta = (X + 1)$ и, обозначив элемент (X) через i, найдем все степени β:

$$\begin{aligned} \beta &= i + 1, \\ \beta^2 &= -i, \\ \beta^3 &= -i + 1, \\ \beta^4 &= -1, \\ \beta^5 &= -i - 1, \\ \beta^6 &= i, \\ \beta^7 &= i - 1, \\ \beta^8 &= 1. \end{aligned}$$

Отсюда видно, что β — примитивный элемент $GF(3^2)$. Используя этот примитивный элемент, найдем минимальные многочлены, соответствующие каждому циклу. Результат будет следующим:

$$\begin{array}{ll} 0: & X - 1 \\ 1, 3: & X^2 + X - 1 \\ 2, 6: & X^2 + 1 \\ 4: & X + 1 \\ 5, 7: & X^2 - X - 1 \end{array}$$

Следовательно, $(X^8 - 1) = (X - 1)(X + 1)(X^2 + 1)(X^2 + X - 1)(X^2 - X - 1)$.

Эта задача дает пример разложения в простом поле [поле называется *простым*, если оно не содержит других полей, кроме самого себя; как мы видели выше, поле может содержать только одно подполе, изоморфное $Z/(p)$, p — простое число (теорема 2.3)]. Остановимся кратко на полях, не являющихся простыми.

Задача 2.62. Поле $GF(q^k)$ является подполем поля $GF(q^j)$ тогда и только тогда, когда $k \mid j$.

Задача 2.63. Разложить двучлен $X^{15} - 1$ в произведение неприводимых над $GF(2^2)$ многочленов.

Решение. Будем считать, что, кроме элементов 0 и 1, поле $GF(2^2)$ содержит также корни β и γ многочлена $X^2 + X + 1$. Элемент β (а следовательно, и γ) является примитивным элементом поля $GF(2^2)$. Представление элементов поля $GF(2^2)$ в виде степеней примитивного элемента будет следующим:

$$\beta = \beta,$$
$$\beta^2 = \beta + 1 + \gamma,$$
$$(\beta^3 = 1).$$

Чтобы разложить $X^{15} - 1$ в произведение неприводимых над $GF(4)$ многочленов, необходимо построить все $(4, 15)$-циклы:

0
1, 4
2, 8
3, 12
5
6, 9
7, 13
10
11, 14

Если сравнить эту таблицу циклов с таблицей $(2, 15)$-циклов из задачи 2.60, то можно заметить, что первая таблица получается разбиением каждого цикла второй таблицы (за исключением цикла 0) на два цикла. Используя примитивный элемент α из задачи 2.60, многочлен, соответствующий $(2, 15)$-циклу, содержащему число 3, можно преобразовать следующим образом:

$$X^4 + X^3 + X^2 + X + 1 = (X - \alpha^3)(X - \alpha^6)(X - \alpha^{12})(X - \alpha^9) =$$
$$= (X - \alpha^3)(X - \alpha^{12}) \cdot (X - \alpha^6)(X - \alpha^9) =$$
$$= (X^2 - (\alpha^3 + \alpha^{12})X + 1)(X^2 - (\alpha^6 + \alpha^9)X + 1).$$

Из вышеизложенного (см. задачу 2.59) имеем $\alpha^3 + \alpha^{12} = \alpha^2 + \alpha + 1 = \alpha^{10}$ и $\alpha^6 + \alpha^9 = \alpha^2 + \alpha = \alpha^5$. Оба элемента α^5 и α^{10} имеют порядок 3 и являются элементами β и γ поля $GF(4)$. Следовательно,

$$X^4 + X^3 + X^2 + X + 1 = (X^2 - \beta X + 1)(X^2 - \gamma X + 1).$$

Аналогично преобразовывая многочлены, соответствующие другим циклам, получим

$$X^{15}-1=(X-1)(X^2-X+\gamma)(X^2-X+\beta)(X^2-\beta X+1)\times$$
$$\times(X^2-\gamma X+1)(X-\gamma)(X^2-\gamma X+\gamma)(X-\beta)(X^2-\beta X+\beta).$$

Выше достаточно подробно было описано, каким образом с помощью таблицы циклов можно находить те или иные минимальные многочлены. Нахождение минимальных многочленов, как мы увидим в следующей главе, является одним из этапов построения циклических кодов. Например, часто приходится выполнять следующую последовательность действий:

1. Выбирается q (q-ичный код).
2. Выбирается n (длина кода).
3. Находится минимальное m, такое, что $n \mid (q^m-1)$, и берется поле $GF(q^m)$.
4. Из элементов поля $GF(q^m)$ выбирается один, имеющий порядок n. Этот элемент обозначается через β и считается фиксированным; (q, n)-циклу, содержащему 1, сопоставляется многочлен $m_\beta(X)$. После этого находятся минимальные многочлены, соответствующие другим интересующим нас циклам.

Задача 2.64. Разложить двучлен $X^{23}-1$ в произведение неприводимых над $GF(2)$ многочленов.

Решение. Построим таблицу $(2, 23)$-циклов:

0

1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12

5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14

Следовательно, двучлен $X^{23}-1$ является произведением $(X-1)$ и двух неприводимых многочленов 11-й степени, двойственных друг другу. Эти неприводимые многочлены в принципе можно найти так же, как и раньше, но такой способ очень сложен. Для нахождения двух сомножителей 11-й степени воспользуемся таблицами, помещенными в приложении. Поскольку порядок корня β двучлена $X^{23}-1$ равен 23, то сначала найдем такие целые числа i, что $23=(2^{11}-1)/(2^{11}-1, i)$. Поскольку, как следует из табл. А.1, $2^{11}-1=23\times 89$, то $i=89$. Если α — примитивный элемент поля $GF(2^{11})$, то α^{89} будет элементом порядка 23. Из табл. А.2 находим, что неприводимым многочленом 11-й степени, имеющим своим корнем элемент α^{89}, является многочлен $X^{11}+X^9+X^7+X^6+X^5+X+1$. Этот многочлен соответствует указанному выше циклу, содержащему 1. Другой ис-

комый многочлен является двойственным к уже найденному многочлену. Таким образом,

$$X^{23}-1=(X+1)(X^{11}+X^9+X^7+X^6+X^5+X+1)(X^{11}+X^{10}+\\+X^6+X^5+X^4+X^2+1).$$

Неприводимые многочлены 11-й степени в правой части последнего разложения являются порождающими многочленами кода Голея.

2.6.* Дополнительные сведения о конечных полях

2.6.1 Вычисления в конечных полях

В конечных полях, как обычно, можно складывать, вычитать, умножать и делить.

Задача 2.65. Решить следующую систему уравнений над $GF(2^4)$:

$$X+\alpha^6Y=\alpha^3,$$
$$\alpha^3X+\alpha^{11}Y=\alpha^4.$$

(Следует воспользоваться таблицей соответствия из задачи 2.59.)

Решение. Используем метод подстановки. Из первого уравнения находим $X=\alpha^3+\alpha^6Y$ и, подставляя его во второе уравнение, получаем

$$(\alpha^9+\alpha^{11})Y=\alpha^4+\alpha^6,$$
$$\alpha^2Y=\alpha^2, \quad \text{откуда} \quad Y=\alpha^{10},$$
$$X=\alpha^3+\alpha=\alpha^9.$$

Для решения указанной выше системы уравнений можно воспользоваться также методом Крамера:

$$X=\frac{\begin{vmatrix}\alpha^3 & \alpha^6\\ \alpha^4 & \alpha^{11}\end{vmatrix}}{\begin{vmatrix}1 & \alpha^6\\ \alpha^3 & \alpha^{11}\end{vmatrix}}=\frac{\alpha^{14}+\alpha^{10}}{\alpha^{11}+\alpha^9}=\frac{\alpha^{11}}{\alpha^2}=\alpha^9,$$

$$Y=\frac{\begin{vmatrix}1 & \alpha^3\\ \alpha^3 & \alpha^4\end{vmatrix}}{\begin{vmatrix}1 & \alpha^6\\ \alpha^3 & \alpha^{11}\end{vmatrix}}=\frac{\alpha^4+\alpha^6}{\alpha^2}=\frac{\alpha^{12}}{\alpha^2}=\alpha^{10}.$$

Утверждение 2.66. *Пусть α — произвольный ненулевой элемент поля $GF(p^m)$. Если существует такой элемент $\beta\in GF(p^m)$, что $\alpha=\beta^2$, то α называется квадратичным вычетом. В против-*

ном случае, т. е. если в $GF(p^m)$ *не существует такого элемента* β, *что* $\alpha = \beta^2$, α *называется квадратичным невычетом.*

1) *Если* $p = 2$, *то любой элемент поля, за исключением элемента* 0, *является квадратичным вычетом.*

2) *Если* p *— нечетное число, то четные степени примитивного элемента являются квадратичными вычетами, а нечетные степени — квадратичными невычетами (следовательно, имеется* $(p^m - 1)/2$ *квадратичных вычетов и столько же квадратичных невычетов).*

2.6.2. Матрицы Адамара

Матрицы Адамара используются в теории планирования экспериментов, теории кодирования, теории связи и других областях. Хорошим введением в теорию матриц Адамара является работа Киясу [8]. Матрицей Адамара H_n порядка n называется квадратная матрица размера $n \times n$ с элементами $+1$ и -1 из поля действительных чисел, такая, что

$$H_n H_n^T = nI,$$

где I — единичная матрица, а H_n^T — матрица, транспонированная к H_n. Например,

$$H_1 = [1],$$

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Утверждение 2.67. *Если* H_n *— матрица Адамара и* $n > 2$, *то* $n \equiv 0 \pmod 4$.

Доказательство. Если $H_n = (a_{ij})$ — матрица Адамара, то $H_n H_n^T = nI$. Это значит, что

$$\sum_{k=1}^{n} a_{ik} a_{jk} = \begin{cases} 0 & \text{при } i \neq j, \\ n & \text{при } i = j. \end{cases}$$

Отсюда следует, что

$$\sum_{k=1}^{n}(a_{1k}+a_{2k})(a_{1k}+a_{3k})=\sum_{k=1}^{n}a_{1k}^2+\sum_{k=1}^{n}a_{1k}a_{2k}+$$
$$+\sum_{k=1}^{n}a_{1k}a_{3k}+\sum_{k=1}^{n}a_{2k}a_{3k}=n.$$

Так как каждое слагаемое в левой части последнего равенства равно 0, 4 или —4, то n кратно четырем.

Высказываются предположения, что существуют матрицы Адамара любого порядка n, кратного четырем. Однако эта гипотеза не доказана и не опровергнута. Для всех $n \leqslant 200$, кратных четырем, за исключением $n = 188$, матрицы Адамара найдены (для этого частично использовались вычислительные машины). Кроме того, известно также несколько конструктивных способов построения матриц Адамара. Один из этих способов дается следующим утверждением.

Утверждение 2.68. *Если существуют матрицы Адамара* H_m *и* H_n, *то существует и матрица Адамара порядка* mn.

Доказательство. Пусть $H_n = (a_{ij})$. Тогда матрица $H_{mn} = (a_{ij}H_m)$ является искомой.

Этим методом можно построить последовательность матриц Адамара, взяв, например, за основу матрицы H_2 и H_4.

Рассмотрим еще один способ построения матриц Адамара, предложенный Палеем. Ряд других методов построения матриц Адамара можно найти также в книге Холла [7].

Пусть q — простое число. Введем функцию[1]) χ, определив ее на множестве элементов поля $GF(q)$ следующим образом:

$$\chi(0)=0,$$
$$\chi(\alpha)=\begin{cases}1: & \alpha\text{ — квадратичный вычет},\\ -1: & \alpha\text{ — квадратичный невычет}.\end{cases}$$

Утверждение 2.69.

1) $\chi(\alpha\beta)=\chi(\alpha)\chi(\beta)$.

2) $\chi(1)=1$.

3) $\sum\limits_{\alpha\in GF(q)}\chi(\alpha)=0$.

4) *Если* $q\equiv 3 \pmod 4$, *то* $\chi(-1)=-1$.

Доказательство. Докажем утверждение 4. Пусть α — примитивный элемент поля $GF(q)$. Тогда

$$\alpha^{q-1}-1=(\alpha^{(q-1)/2}-1)(\alpha^{(q-1)/2}+1)=0.$$

[1]) χ называется символом Лежандра; см. также разд. 8.1.9. — *Прим. ред.*

Однако так как все элементы $1, \alpha, \alpha^2, \ldots, \alpha^{q-2}$ различны, то $\alpha^{(q-1)/2} \neq 1$. Следовательно,

$$\alpha^{(q-1)/2} = -1.$$

Но тогда

$$\chi(-1) = \begin{cases} +1, & q = 4t+1, \\ -1, & q = 4t-1, \end{cases} \quad \text{т. е.} \quad q \equiv 3 \bmod 4.$$

Утверждение 2.70. *Если* $\beta \neq 0$, *то*

$$\sum_{\alpha \in GF(q)} \chi(\alpha)\chi(\alpha+\beta) = -1.$$

Доказательство. Если $\alpha = 0$, то $\chi(0)\chi(0+\beta) = 0$. Допустим, что $\alpha \neq 0$. Тогда в поле $GF(q)$ найдется, и притом единственный, элемент γ, такой, что $\alpha + \beta = \alpha\gamma$. Когда α пробегает все ненулевые элементы поля $GF(q)$, γ пробегает также все ненулевые элементы поля $GF(q)$, за исключением 1. Следовательно,

$$\sum_{\alpha} \chi(\alpha)\chi(\alpha+\beta) = \sum_{\alpha \neq 0} \chi(\alpha)\chi(\alpha+\beta) = \sum_{\alpha \neq 0} \chi(\alpha^2)\chi(\gamma) =$$

$$= \sum_{\gamma \neq 1} \chi(\gamma) = \sum_{\gamma} \chi(\gamma) - \chi(1) = 0 - 1 = -1.$$

Используя приведенные выше утверждения и отождествляя элементы поля $GF(q)$ с вычетами $1, \ldots, q-1, q(\equiv 0)$, по модулю q, построим следующую квадратную матрицу $H = (a_{ij})$ порядка $q+1$:

$$a_{i0} = a_{0j} = 1, \quad 0 < i,\ j \leqslant q,$$

$$a_{ii} = -1, \quad 0 \leqslant i \leqslant q,$$

$$a_{ij} = \chi(j-i), \quad 0 < i,\ j \leqslant q, \quad i \neq j.$$

Утверждение 2.71. *Если* q — *простое число, такое, что* $q \equiv 3 (\mathrm{mod}\ 4)$, *то построенная выше матрица* H *является матрицей Адамара.*

Доказательство. Нужно показать, что $HH^T = (q+1)I$. Очевидно, что диагональные элементы b_{ij} матрицы $B = HH^T = (b_{ij})$ равны $q+1$. Из части 3 утверждения 2.69 непосредственно следует, что

$$b_{i0} = b_{0j} = 0, \quad 0 < i,\ j \leqslant q.$$

Покажем, что все остальные элементы b_{ij} $(0 < i, j \leqslant q, i \neq j)$ матрицы B равны нулю Имеем

$$b_{ij} = \sum_{k=0}^{q} a_{ik}a_{jk} = 1 + \sum_{k=1}^{q} \chi(k-i)\chi(k-j) + (-1)\chi(i-j) +$$

$$+ (-1)\chi(j-i).$$

Положим $l = k - i$ и $\beta = i - j$. Когда k пробегает все значения от 1 до q, то l также пробегает все элементы поля $FG(q)$. Заметим, что в данном случае $\beta \neq 0$. Кроме того, согласно утверждению 2.69, $(-1)\chi(j-i) = \chi(-1)\chi(j-i) = \chi(i-j)$. Тогда из утверждения 2.70 следует, что

$$b_{ij} = 1 + \sum_{l \in GF(q)} \chi(l)\chi(l+\beta) = 1 - 1 = 0.$$

Как известно (см., например, [4]), существует бесконечная последовательность простых чисел q, таких, что $q \equiv 3 \pmod 4$. Это означает, что описанная выше конструкция дает бесконечный класс матриц Адамара.

С помощью матриц Адамара можно строить некоторые двоичные коды. Один из таких классов кодов — это эквидистантные коды, описанные в разд. 1.4.2. Другой класс кодов, которые могут быть построены с помощью матриц Адамара, — это коды Рида — Маллера, рассматриваемые в разд. 4.5.1.

Утверждение 2.72.

1) *При одновременной перестановке строк и столбцов, а также при умножении строк и столбцов матрицы Адамара на* -1 *опять получается матрица Адамара. В случае необходимости с помощью операций этих двух типов всегда можно получить приведенную матрицу Адамара, все элементы первой строки и первого столбца которой равны* $+1$. *Рассмотрим матрицу размера* $n \times (n-1)$, *получающуюся из приведенной матрицы Адамара порядка* n *удалением первого столбца. Заменим в этой матрице элементы* $+1$ *на* 0, *а* -1 *и каждую строку получающейся матрицы будем рассматривать как кодовое слово блокового кода длины* $n-1$. *Совокупность таких* n *кодовых слов представляет собой эквидистантный код. Заметим, что этот код, вообще говоря, не является линейным.*

2) *Рассмотрим совокупность из* $2n$ *векторов, первые* n *из которых* $v_1, \ldots, v_n$ *являются строками матрицы Адамара порядка* n, *а вторые* n *получаются умножением первых на* -1. *Аналогично тому, как это мы сделали выше, заменим элементы* $+1$ *на* 0, *а элементы* -1 *на* 1. *Полученная таким образом совокупность из* $2n$ *двоичных векторов представляет собой блоковый код длины* n *с кодовым расстоянием* $n/2$. *Вообще говоря, этот код также не является линейным. Однако коды этого типа, построенные на основе матриц Адамара* H_{2^m}, *являются не чем иным, как кодами Рида — Маллера первого порядка. Существование матриц Адамара* H_{2^m} *для любого целого* m *следует из утверждения* 2.68.

2.6.3. Конечные геометрии

В этом разделе рассматриваются главным образом конечные евклидовы геометрии на языке их конечных представлений. Конечные геометрии так же, как группы и поля, являются системами, которые определяются совокупностью аксиом, однако в данном случае вполне можно отказаться от общности изложения и рассмотреть конечные геометрии в одном из их конкретных представлений. Детальное изложение теории конечных геометрий можно найти в [5]. Конечные геометрии используются в теории планирования экспериментов [6], в теории кодирования для введения проективно-геометрических кодов и для других целей.

Конечные евклидовы геометрии. Последовательности из m элементов поля $GF(q)$

$$X=(x_1, x_2, \ldots, x_m), \quad x_i \in GF(q),$$

будем называть далее точками. Пусть $X=(x_1, \ldots, x_m)$ и $Y=(y_1, \ldots, y_m)$ — две различные точки. Совокупность точек, где α — элемент поля $GF(q)$,

$$\alpha X+(1-\alpha)Y=(\alpha x_1+(1-\alpha)y_1, \ldots, \alpha x_m+(1-\alpha)y_m, \ \alpha \in GF(q),$$

назовем прямой, проходящей через X и Y, и обозначим через $X \vee Y$. Следовательно, одна прямая содержит q точек. Множество из q^m точек и систему прямых, определенных указанным выше способом, назовем *евклидовой геометрией* размерности m и будем обозначать через $EG(m, q)$. Параметр q евклидовой геометрии $GF(m, q)$ указывает не на то, что последняя «берется над полем $GF(q)$», а определяет число точек, содержащихся в одной прямой. Использованный здесь способ описания конечной евклидовой геометрии, при котором каждая точка задается своими координатами, являющимися элементами поля $GF(q)$, представляет собой не более чем один из способов представления $EG(m, q)$. Некоторые авторы обозначают введенную выше конструкцию не через $EG(m, q)$, а через $AG(m, q)$. При этом они исходят из того, что геометрию $EG(m, q)$, в которой не введено расстояние, следует называть *аффинной геометрией.*

Определим k-пространство в $EG(m, q)$ как множество из q^k точек $X=(x_1, \ldots, x_m)$, удовлетворяющих соотношению

$$AX^T=b^T,$$

где X^T и b^T — вектор-столбцы, полученные транспонированием последовательностей X и $b=(b_1, \ldots, b_{m-k})$, а A — матрица размера $(m-k) \times m$ с элементами из поля $GF(q)$, имеющая ранг $m-k$. В частности, 0-пространство называется точкой, 1-пространство — прямой, 2-пространство — плоскостью, а $(m-1)$-пространство — *гиперплоскостью* в $EG(m, q)$. Другими словами,

гиперплоскость — это множество решений $(x_1, \ldots, x_m)$ одного линейного уравнениия

$$a_1x_1 + a_2x_2 + \ldots + a_mx_m = b.$$

Поэтому k-пространство — это пересечение $m - k$ гиперплоскостей. Вначале прямая была определена как совокупность линейных комбинаций двух точек, а затем прямой было названо также 1-пространство. Покажем, что в действительности оба эти определения эквивалентны. Пусть $X = (x_1, \ldots, x_m)$ и $Y = (y_1, \ldots, y_m)$ — две различные точки, принадлежащие 1-пространству. Рассмотрим все линейные комбинации $\alpha X + (1 - \alpha) Y$ этих двух точек. Так как $AX^T = b^T$ и $AY^T = b^T$, то

$$A[\alpha X + (1 - \alpha) Y]^T = \alpha b^T + (1 - \alpha) b^T = b^T,$$

т. е. все линейные комбинации X и Y принадлежат 1-пространству. Далее рассмотрим совокупность точек $Z = (z_1, \ldots, z_m)$, являющихся линейными комбинациями $\alpha X + (1 - \alpha) Y$ точек X и Y. В этом случае

$$\begin{aligned} z_1 &= \alpha x_1 + (1 - \alpha) y_1, \\ &\vdots \\ z_m &= \alpha x_m + (1 - \alpha) y_m. \end{aligned}$$

Отсюда следует, что

$$(x_2 - y_2) z_1 - (x_1 - y_1) z_2 = y_1x_2 - x_1y_2.$$

Таким образом можно получить $m - 1$ независимых соотношений, связывающих координаты $z_1, \ldots, z_m$ точек Z.

Сформулированные выше утверждения могут быть распространены на произвольные k-пространства. Действительно, рассмотрим $k + 1$ точек $X_0 = (x_{01}, \ldots, x_{0m}), \ldots, X_k = (x_{k1}, \ldots, x_{km})$, таких, что ранг матрицы $X = (x_{ij})$ равен k; такие точки называются линейно независимыми. Определим k-пространство как множество из q^k точек следующего вида:

$$\sum_{i=0}^{k} \alpha_i X_i, \quad \sum \alpha_i = 1, \quad \alpha_i \in GF(q).$$

Это определение k-пространства эквивалентно приведенному выше; k-пространство, порождаемое $k + 1$ независимыми точками $X_0, X_1, \ldots, X_k$, будем обозначать через $X_0 \vee X_1 \vee \ldots \vee X_k$.

Задача 2.73. Найти все 2-пространства в $EG(3, 2)$.

Решение. В данном случае 2-пространства являются гиперплоскостями и задаются одним линейным уравнением. С другой

стороны, выбор трех линейно независимых точек X, Y и Z также задает 2-пространство, которое в этом случае представляет собой множество из четырех точек $\{X, Y, Z, X+Y+Z\}$. Всего имеется 14 различных 2-пространств:

	$b=0$	$b=1$
$x_1=b$	(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1)	(1, 0, 0), (1, 0, 1) (1, 1, 0), (1, 1, 1)
$x_2=b$	(0, 0, 0), (0, 0, 1), (1, 0, 0), (1, 0, 1)	(0. 1, 0), (0, 1, 1), (1. 1, 0), (1, 1, 1)
$x_3=b$	(0, 0. 0), (0, 1, 0), (1, 0, 0), (1, 1, 0)	(0, 0, 1), (0, 1, 1), (1, 0, 1), (1, 1, 1)
$x_1+x_2=b$	(0, 0, 0), (0, 0, 1), (1, 1, 0), (1, 1, 1)	(0, 1. 0), (0, 1, 1), (1, 0, 0), (1, 0, 1)
$x_1+x_3=b$	(0, 0, 0), (0, 1, 0), (1, 0, 1), (1. 1, 1)	(0, 0, 1), (0, 1, 1), (1, 0, 0), (1, 1, 0)
$x_2+x_3=b$	(0, 0, 0), (1, 0, 0), (0, 1, 1), (1, 1, 1)	(0, 0, 1), (1, 0. 1), (0, 1, 0), (1. 1, 0)
$x_1+x_2+x_3=b$	(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)	(0, 0, 1), (0, 1, 0), (1, 0, 0), (1, 1, 1)

Мы установили, что в $EG(3, 2)$ существует 14 различных 2-пространств. Попробуем найти число $W(m, k, q)$ различных k-пространств в $EG(m, q)$ в общем случае.

Утверждение 2.74. *Число $W(m, k, q)$ различных k-пространств в $EG(m, q)$ определяется следующей формулой*:

$$W(m, k, q) = q^{m-k}\frac{(q^m-1)\ldots(q^{m-k+1}-1)}{(q^k-1)\ldots(q-1)}.$$

Доказательство. Вначале определим число способов выбора точек $X_0, \ldots, X_k$, порождающих k-пространство $X_0 \vee X_1 \vee \ldots \vee X_k$. Из множества всех q^k точек точку X_0 можно выбрать q^m способами. Исключив из множества всех q^m точек точку X_0, точку X_1 можно выбрать $q^m - 1$ способами. Так как точка X_2 не должна входить в $X_0 \vee X_1$, то ее можно выбрать $q^m - q$ способами и т. д. Таким образом, число способов выбора точек $X_0, \ldots, X_k$ равно

$$q^m(q^m-1)(q^m-q)\ldots(q^m-q^{k-1}).$$

Однако $X_0 \vee X_1 \vee \ldots \vee X_k = X_1 \vee X_0 \vee \ldots \vee X_k = \ldots$. Эти равенства показывают, что одно и то же k-пространство имеет различные представления. Аналогично тому, как выше было подсчитано число способов выбора $X_0, \ldots, X_k$, можно найти также и число способов представления одного фиксированного k-пространства. Это число оказывается равным

$$q^k(q^k-1)(q^k-q)\ldots(q^k-q^{k-1}).$$

Следовательно, искомое число различных k-пространств определяется равенством

$$W(m, k, q) = q^{m-k}\frac{(q^m-1)(q^m-q)\ldots(q^m-q^{k-1})}{(q^k-1)(q^k-q)\ldots(q^k-q^{k-1})}.$$

Как показывает пример евклидовой геометрии $EG(3, 2)$, пересечение двух 2-пространств является либо прямой, либо пустым множеством. Для доказательства утверждения 2.76 потребуется следующее обобщение этого факта.

Утверждение 2.75. *Два $(k+1)$-пространства, содержащие некоторое k-пространство, не имеют общих точек, отличных от точек этого k-пространства.*

Доказательство. Пусть $X_1, \ldots, X_{k+1}$ — линейно независимые точки, порождающие общее k-пространство C, и пусть D_1 и D_2 — два различных $(k+1)$-пространства, содержащих C. Предположим, что D_1 и D_2 имеют общую точку X_0, которая не входит в C. Тогда точки $X_0, X_1, \ldots, X_{k+1}$ задают одно $(k+1)$-пространство, т. е. $D_1 = D_2$, и мы получили противоречие.

Утверждение 2.76. *Число различных $(k+1)$-пространств в $EG(m, q)$, содержащих заданное k-пространство, равно*

$$\frac{q^{m-k}-1}{q-1}.$$

Доказательство. Число точек, не принадлежащих рассматриваемому k-пространству C, равно $q^m - q^k$. Любое $(k+1)$-пространство задается $k+1$ линейно независимыми точками, одна из которых не принадлежит C, а остальные k входят в C. Число точек $(k+1)$-пространства, не являющихся точками C, равно $q^{k+1} - q^k$. Пусть J — число $(k+1)$-пространств, содержащих C. Тогда, принимая во внимание утверждение 2.75, получаем

$$J(q^{k+1} - q^k) = q^m - q^k.$$

Конечные проективные геометрии. Как и в случае евклидовых геометрий, будем представлять точки (а точнее объекты, называемые точками) посредством координат. *Конечную проективную геометрию*, каждая прямая которой состоит из $q+1$ точек, будем обозначать через $PG(m, q)$. Конечная проективная геометрия $PG(m, q)$ состоит из конечного числа точек, каждая из которых может быть представлена своими координатами:

$$X = (x_0, x_1, \ldots, x_m), \quad x_i \in GF(q).$$

Однако в случае проективной геометрии это представление точки не однозначно, а именно считается, что последовательности $X = (x_0, x_1, \ldots, x_m)$ и $\alpha X = (\alpha x_0, \ldots, \alpha x_m)$, $\alpha \in GF(q) - \{0\}$, определяют одну и ту же точку. Следовательно, каждая точка в $PG(m, q)$ может быть представлена $q-1$ способами. Последо-

вательность (0, ..., 0) в данном случае из рассмотрения исключается. Таким образом, число точек в $PG(m, q)$ равно

$$\frac{q^{m+1}-1}{q-1}.$$

Пусть $X=(x_0, \ldots, x_m)$ и $Y=(y_0, \ldots, y_m)$ — представления двух различных точек. Множество точек, которые могут быть представлены в виде

$$\alpha X+\beta Y=(\alpha x_0+\beta y_0, \ldots, \alpha x_m+\beta y_m)$$

[$\alpha, \beta \in GF(q)$, сочетание $\alpha=\beta=0$ исключается], называется прямой, проходящей через точки X и Y (или *1*-пространством) и обозначается через $X \vee Y$. Поскольку пара (α, β) может быть выбрана q^2-1 различными способами, то одна прямая состоит из

$$\frac{q^2-1}{q-1}=q+1$$

точек, на что и указывает параметр q в обозначении $PG(k, q)$ проективной геометрии.

Пусть $Z=(z_0, \ldots, x_m)$ — точка, не лежащая на прямой $X \vee Y$. Множество всех точек вида

$$\alpha X+\beta Y+\gamma Z,$$

где α, β и γ — не равные одновременно нулю элементы поля $GF(q)$, называется плоскостью (или 2-пространством), проходящей через точки X, Y и Z, и обозначается через $X \vee Y \vee Z$. Каждая плоскость состоит из

$$\frac{q^3-1}{q-1}=q^2+q+1$$

точек.

Задача 2.77. Представьте $PG(2, 2)$ в указанном выше виде.

Решение. Так как размерность $PG(2, 2)$ равна 2, то $PG(2, 2)$ представляет собой проективную плоскость, состоящую из $(2^3-1)/(2-1)=7$ точек. В качестве представлений этих семи точек могут быть взяты следующие последовательности [1]):

(0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0,1), (1, 1, 0), (1, 1, 1).

В $PG(2, 2)$ имеется семь прямых, каждая из которых состоит из трех точек:

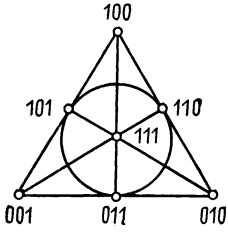
{(0, 0, 1), (0, 1, 0), (0, 1, 1)}, {(0, 1, 0), (1, 0, 0), (1, 1, 0)}, {(0, 1, 1), (1, 0, 1), (1, 1, 0)}, {(0, 0, 1), (1, 0, 0), (1, 0, 1)}, {(0, 1, 0), (1, 0, 1), (1, 1, 1)}, {(0, 0, 1), (1, 1, 0), (1, 1, 1)}, {(0, 1, 1), (1, 0, 0), (1, 1, 1)}.

[1]) Поскольку $GF(2)$ — {0} состоит из одного элемента, любая точка представляется в данном случае однозначно. — *Прим. ред.*

На фиг. 2.2 показано расположение точек и прямых $PG(2, 2)$ на плоскости.

Легко видеть, что проективная плоскость $PG(2, q)$ существует, если q является простым числом или степенью простого числа. Если же q — составное число, не являющееся степенью простого числа, то о существовании проективной плоскости $PG(2, q)$ в общем случае ничего сказать нельзя. Известно, что проективная плоскость $PG(2, 6)$ не существует, а вопрос о существовании $PG(2, 10)$ является открытым.

В $PG(m, q)$ точно так же, как в $EG(m, q)$, k-пространство может быть определено и как пересечение гиперплоскостей, и



Фиг. 2.2.

как совокупность линейных комбинаций $(k+1)$ линейно независимых точек. При этом гиперплоскость определяется как множество решений $(x_0, \ldots, x_m)$ следующего уравнения первой степени:

$$a_0x_0 + a_1x_1 + \ldots + a_mx_m = 0$$

(тривиальное решение $(0, \ldots, 0)$ исключается). Второе из указанных выше определений k-пространства более строго может быть сформулировано следующим образом: k-пространство — это совокупность всех линейных комбинаций $(k+1)$ линейно независимых точек $X_0, \ldots, X_m$ следующего вида:

$$\alpha_0X_0 + \alpha_1X_1 + \ldots + \alpha_kX_k,$$

где $\alpha_0, \alpha_1, \ldots, \alpha_k$ — элементы поля $GF(q)$, одновременно не равные нулю. Оба эти определения полностью эквивалентны.

Утверждение 2.78. *Число $V(m, k, q)$ k-пространств в $PG(m, q)$ определяется формулой*

$$V(m, k, q) = \frac{(q^{m+1}-1)(q^m-1)\ldots(q^{m+1-k}-1)}{(q^{k+1}-1)(q^k-1)\ldots(q-1)}.$$

Между $PG(m, q)$ и $EG(m, q)$ существует тесная связь. Сопоставим точке $X = (x_1, \ldots, x_m)$ из $EG(m, q)$ точку из $PG(m, q)$

по следующему правилу:

$$(x_1, \ldots, x_m) \to (1, x_1, \ldots, x_m).$$

Обратное отображение при $x_0 \neq 0$ задается формулой

$$(x_0, x_1, \ldots, x_m) \to (x_1/x_0, \ldots, x_m/x_0).$$

Точки в $PG(m, q)$ с $x_0 = 0$ называются *бесконечно удаленными.* При описанном выше соответствии k-пространства переходят в k-пространства.

2.6.4. Разностные множества

Множество из k целых чисел $(d_1, \ldots, d_k)$ называется (совершенным) *разностным* (v, k, λ)-множеством, если при делении разностей $d_i - d_j$, $i \neq j$, на целое число v каждый из остатков $1, 2, \ldots, v-1$ встречается ровно λ раз.

Из определения непосредственно следует, что

$$k(k-1) = \lambda(v-1). \tag{2.15}$$

Утверждение 2.79. *Пусть* $(d_1, \ldots, d_k)$ *— разностное* $(v, k, 1)$*-множество,* $P = \{0, 1, \ldots, v-1\}$ *и*

$$L_j = \{j + d_i \pmod v;\ i = 1, \ldots, k\}, \quad j = 0, 1, \ldots, v-1.$$

Тогда множество P*, рассматриваемое как множество точек, и система прямых* $L_0, \ldots, L_{v-1}$ *являются конечной проективной геометрией размерности* $k-1$.

Пример. Множество (0, 1, 3) является разностным (7, 3, 1)-множеством.

$$P = \{0, 1, 2, 3, 4, 5, 6\},$$
$$L_0 = \{0, 1, 3\}, \quad L_1 = \{1, 2, 4\}, \quad L_2 = \{2, 3, 5\}, \quad L_3 = \{3, 4, 6\},$$
$$L_4 = \{4, 5, 0\}, \quad L_5 = \{5, 6, 1\}, \quad L_6 = \{6, 0, 2\}.$$

Получающаяся из этого разностного множества проективная геометрия совпадает с рассмотренной выше проективной геометрией $PG(2, 2)$.

Рассмотрим один из способов построения разностных множеств. Заметим, что, кроме этого способа существует много других способов, подробное описание которых можно найти в [7].

Утверждение 2.80. *Пусть* $v = 4\lambda + 3$ *— простое число, такое, что* $v \equiv 3 \pmod 4$. *Тогда множество* D_λ *квадратичных вычетов по модулю* v *является разностным* $(4\lambda + 3, 2\lambda + 1, \lambda)$*-множеством.*

Доказательство. Нужно показать, что при любом d $(0 < d < v)$ сравнение $x^2 - y^2 \equiv d \pmod v$, $0 < x, y < v$, имеет одно и то же число решений (x, y). Предположим, что сравнение $x^2 - y^2 \equiv 1 \pmod v$ имеет μ решений $(a_1, b_1), \ldots, (a_\mu, b_\mu)$, $0 < a_i, b_i < v$. Если d — квадратичный вычет, то $d \equiv e^2$ и решениями сравнения $x^2 - y^2 \equiv d$ являются $(ea_1, eb_1), \ldots, (ea_\mu, eb_\mu)$, Если d — квадратичный невычет, то, согласно утверждению 2.69 (4), квадратичным вычетом является $-d$. Полагая $-d \equiv e^2$, получаем, что решениями сравнения $x^2 - y^2 \equiv d$ являются $(eb_1, ea_1), \ldots, (eb_\mu, ea_\mu)$. Таким образом, мы доказали, что для каждого d сравнение $x^2 - y^2 \equiv d \pmod v$ имеет ровно μ решений, а это означает, что D_λ — разностное множество. Как следует из формулы (2.14), $\mu = \lambda$.

Пример. $\lambda = 2$, $v = 11$, $k = 5$. Так как в этом случае $1^2 \equiv 1$, $2^2 \equiv 4$, $3^2 \equiv 9$, $4^2 \equiv 5$, $5^2 \equiv 3$, $6^2 \equiv 3$, $7^2 \equiv 5$, $8^2 \equiv 9$, $9^2 \equiv 4$, $10^2 \equiv 1$, то $D_2 = \{1, 3, 4, 5, 9\}$. Если найти все разности, то можно непосредственно убедиться, что каждое из чисел, отличное от 0 и меньше 11, входит в множество разностей ровно 2 раза. Действительно,

—	1	3	4	5	9
1	0	9	8	7	3
3	2	0	10	9	5
4	3	1	0	10	6
5	4	2	1	0	7
9	8	6	5	4	0

2.6.5. Дополняющий базис

Пусть $E = GF(q)$, $q = p^t$, p — простое число. *След* $T_q(\lambda)$ элемента $\lambda \in GF(q^m)$ относительно E определяется соотношением

$$T_q(\lambda) = \lambda + \lambda^q + \lambda^{q^2} + \ldots + \lambda^{q^{m-1}}.$$

Утверждение 2.81. *Для любого базиса* $\alpha_1, \ldots, \alpha_m$ *векторного пространства* $GF(q^m)$ *над полем* E *существует двойственный базис* $\lambda_1, \ldots, \lambda_m$, *обладающий следующим свойством*:

$$T_q(\lambda_j \alpha_j) = \begin{cases} 0, & i \neq j, \\ 1, & i = j. \end{cases}$$

Доказательство. Рассмотрим матрицу

$$A = \begin{bmatrix} \alpha_1 & \alpha_1^q & \dots & \alpha_1^{q^{m-1}} \\ \alpha_2 & \alpha_2^q & \dots & \alpha_2^{q^{m-1}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \alpha_m & \alpha_m^q & \dots & \alpha_m^{q^{m-1}} \end{bmatrix}.$$

Матрица A является *невырожденной*, поскольку элементы $\alpha_1, \dots, \alpha_m$ линейно независимы над E. Следовательно, матрица A имеет обратную матрицу A^{-1}. *Определитель* A обозначим через $|A|$. Пусть $M = (a_{ij})$ — квадратная матрица порядка n, и пусть A_{ij} — определитель матрицы, получающейся из матрицы M удалением i-й строки и j-го столбца; A_{ij} называется *минором*. При этом величина $(-1)^{i+j}\Delta_{ij}$ называется *алгебраическим дополнением* элемента a_{ij} и обозначается через A_{ij}. Матрица

$$\begin{bmatrix} A_{11} & A_{21} & \dots A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{bmatrix}$$

называется *дополнительной матрицей* матрицы M и обозначается через adj M. Матрица adj M обладает следующим свойством:

$$(\operatorname{adj} M) \cdot M = M \cdot (\operatorname{adj} M) = |M| \cdot I_n,$$

где I_n — единичная матрица порядка n.

Попробуем найти матрицу adj A:

$$A_{11} = \begin{vmatrix} \alpha_2^q & \dots & \alpha_2^{q^{m-1}} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \alpha_m^q & \dots & \alpha_m^{q^{m-1}} \end{vmatrix}, \quad A_{12} = \begin{vmatrix} \alpha_2 & \alpha_2^{q^2} & \dots & \alpha_2^{q^{m-1}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \alpha_m & \alpha_m^{q^2} & \dots & \alpha_m^{q^{m-1}} \end{vmatrix}$$

и т. д. Имеют место также следующие равенства:

$$A_{12} = (-1)^{m'} A_{11}^q, \quad A_{13} = A_{11}^{q^2}, \quad A_{14} = (-1)^{m'} A_{11}^{q^3}, \dots,$$
$$A_{22} = (-1)^{m'} A_{12}^q, \quad A_{23} = A_{12}^{q^2}, \quad A_{24} = (-1)^{m'} A_{12}^{q^3}, \dots,$$
$$\dots\dots\dots\dots\dots\dots\dots\dots$$

где $m' = m - 1$. Докажем, например, справедливость равенства $A_{12} = (-1)^{m'} A_{11}^q$. Согласно определению определителя,

$$A_{11} = \sum \sigma(p_1, \dots, p_m) a_{1p_1} a_{2p_2} \dots a_{mp_m},$$

где сумма берется по всем наборам чисел $(p_1, \ldots, p_m)$, получающимся в результате перестановки чисел $(1, 2, \ldots, m)$, а $\sigma(p_1, \ldots, p_m)$ — знак перестановки.

Воспользовавшись вышеизложенным (см. задачу 2.27), получим

$$A_{11}^q = \sum \sigma(p_1, \ldots, p_m)\, a_{1p_1}^q \ldots a_{mp_m}^q =$$

$$= \begin{vmatrix} \alpha_2^{q^2} & \ldots & \alpha_2^{q^{m-1}} & \alpha_2 \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot \\ \alpha_m^{q^2} & \ldots & \alpha_m^{q^{m-1}} & \alpha_m \end{vmatrix} = (-1)^{m'} \begin{vmatrix} \alpha_2 & \alpha_2^{q^2} & \ldots & \alpha_2^{q^{m-1}} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \alpha_m & \alpha_m^{q^2} & \ldots & \alpha_m^{q^{m-1}} \end{vmatrix} =$$

$$= (-1)^{m'} A_{12}.$$

Аналогично доказываются и остальные приведенные выше равенства. Из этих равенств следует, что $|A|^q = (-1)^{m'} |A|$. Следовательно, первым столбцом матрицы $A^{-1} = \operatorname{adj} A / |A|$ является вектор

$$(A_{11}/|A|,\ (-1)^{m'} A_{11}^q/|A|,\ A_{11}^{q^2}/|A|, \ldots)^T =$$

$$= (A_{11}/|A|,\ A_{11}^q/|A|^q,\ A_{11}^{q^2}/|A|^{q^2}, \ldots)^T =$$

$$= (\lambda_1, \lambda_1^q, \lambda_1^{q^2}, \ldots, \lambda_1^{q^{m-1}})^T$$

где $\lambda_1 = A_{11}/|A|$. Аналогично через $\lambda_i = A_{1i}/|A|$ можно выразить и остальные столбцы матрицы A^{-1}. Таким образом,

$$A^{-1} = \begin{vmatrix} \lambda_1 & \ldots & \lambda_m \\ \lambda_1^q & \ldots & \lambda_m^q \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \lambda_1^{q^{m-1}} & \ldots & \lambda_m^{q^{m-1}} \end{vmatrix}.$$

Очевидно, что элементы $\lambda_1, \ldots, \lambda_m$ линейно независимы (если допустить, что они линейно зависимы, то должны существовать такие элементы $c_i \in E$, что $\sum c_i \lambda_i = 0$ и $(c_1, \ldots, c_m) \neq (0, \ldots, 0)$). В этом случае $\sum c_i^q \lambda_i^q = \sum c_i \lambda_i^q = 0$ и $\sum c_i \lambda_i^{q^j} = 0$, $j = 1, 2, \ldots, m-1$, но это противоречит тому, что A^{-1} — невырожденная матрица).

Из равенства $AA^{-1} = I_n$ получаем

$$\sum_{i=1}^{m} \lambda_i^{q^j} \alpha_i^{q^j} = 1, \quad \sum_{i=1}^{m} \lambda_i^{q^j} \alpha_i^{q^k} = 0 \quad (j \neq k).$$

Утверждение 2.81 доказано.

2.6.6. Некоторые понятия, необходимые для определения кодов Гоппы

Пусть L — некоторое подмножество элементов поля $GF(2^m)$. Элементы множества L обозначим через $\alpha_1, \ldots, \alpha_n$, $n \leqslant 2^m$. Пусть S — векторное пространство размерности n над $GF(2)$. Каждому вектору $x = (a_1, \ldots, a_n) \in S$ поставим в соответствие рациональную функцию

$$R_x(z) = \sum_{i=1}^{n} \frac{a_i}{z - \alpha_i}.$$

Это отображение является гомоморфным отображением S в аддитивную группу рациональных функций над $GF(2^m)$.

Если вектору x, у которого $a_{i_1} = a_{i_2} = \ldots = a_{i_k} = 1$, а остальные компоненты равны 0, сопоставить многочлен $f(z) = (z - \alpha_{i_1}) \ldots (z - \alpha_{i_k})$, то $R_x(z)$ можно представить в виде

$$R_x(z) = \frac{f'(z)}{f(z)},$$

где $f'(z)$ — формальная производная $f(z)$ (см. разд. 2.4).

Рассмотрим один из многочленов $g(z)$ с коэффициентами из поля $GF(2^m)$, среди корней которого нет элементов множества L. Имеет место следующее соотношение:

$$R_x(z) = \sum_{i=1}^{n} \frac{a_i}{z - \alpha_i} \equiv 0 \pmod{g(z)} \Leftrightarrow \sum_{i=1}^{n} a_i \{(z - \alpha_i)^{-1}\}_m = 0, \qquad (2.16)$$

где $\{(z - \alpha_i)^{-1}\}_m$ — элемент кольца многочленов по модулю $g(z)$, обратный к $(z - \alpha_i)$, который имеет следующий вид:

$$\{(z - \alpha_i)^{-1}\}_m \equiv \frac{g(z) - g(\alpha_i)}{z - \alpha_i} g^{-1}(\alpha_i). \qquad (2.17)$$

Действительно, $\{(z - \alpha_i)^{-1}\}_m (z - \alpha_i) = g^{-1}(\alpha_i) g(z) - 1 \equiv -1 \pmod{g(z)}$. В справедливости (2.16) можно убедиться, если заметить, что

$$\frac{1}{z - \alpha_i} \equiv \frac{g(z) - g(\alpha_i)}{z - \alpha_i} g^{-1}(\alpha_i) \pmod{g(z)}.$$

[Последнее соотношение имеет место потому, что степень многочлена в правой части (2.17), меньше, чем степень $g(z)$.]

Задачи

1. (Группы.) Привести примеры групп порядка $n \leqslant 3$ (построить таблицы групповой операции). Построить таблицы групповых операций всех неизоморфных групп порядка $n = 4$ (сле-

дует воспользоваться пояснениями, приведенными перед задачей 2.4).

2. (Конечные поля). Найти значения $(-1), (-1)^2, (-1)^3, \ldots$ в поле характеристики p (p — простое число) (к задаче 2.27).

3. (Конечные поля.) Решить уравнение второго порядка над $GF(2)$: $aX^2 + bX + c = 0$. В случае если оно не имеет решений в поле $GF(2)$, найти корни, расширив соответствующим образом поле $GF(2)$.

4. (Неприводимые многочлены.) Ниже указаны неприводимые над $Z/(3)$ многочлены второй, третьей и четвертой степеней:
второй степени: $X^2 + 1$, $X^2 + X + 2$, $X^2 + 2X + 2$,
третьей степени: $X^3 + 2X + 1$,
четвертой степени: $X^4 + X^3 + X^2 + 2X + 2$.
Доказать, что эти многочлены неприводимы. Каждый элемент (i) поля $Z/(3)$ для простоты можно обозначать через i (к задачам 2.43 и 2.44).

5. (Построение конечных полей.) Построить поле $GF(3^2)$, используя для этого неприводимый многочлен $X^3 + 2X + 1$ (к задаче 2.51).

6. (Представление конечных полей.) Поле $GF(2^4)$ можно построить как с помощью неприводимого многочлена $X^3 + X + 1$, так и с помощью неприводимого многочлена $X^3 + X^2 + X + 1$. Получить таблицу соответствия представлений элементов (к теореме 2.12).

7. (Примитивный элемент.) Попробуйте найти примитивные элементы поля $Z/(p)$ простого порядка в случае, если $p = 3, 5, 7, 11, 13, 17, 19, 23$ (при $p = 7$ имеются два примитивных элемента 3 и 5, утверждение 2.57).

8. (Конечные проективные геометрии.) Исследовать структуру $PG(4, 3)$ (определить числа точек, прямых, плоскостей, 3-пространств; определить числа точек, входящих в указанные множества).

9. (Разностные множества.) Построить описанным в утверждении 2.80 методом разностные множества в случаях $\lambda = 3$ и $\lambda = 4$.

10. (Условие неприводимости многочлена степени 2.) Пусть β — элемент $GF(q^m)$. След элемента β относительно $GF(q)$ был определен следующим образом:

$$T_q(\beta) = \sum_{i=1}^{m-1} \beta^{q^i}.$$

Если $\beta \in GF(2^m)$ и $T_2(\beta) \neq 0$, то многочлен $X^2 + X + \beta$ является неприводимым над $GF(2^m)$. Это можно доказать следующим образом. Пусть

$$X^2 + X + \beta = (X + \beta_1)(X + \beta_2), \quad \beta_1, \beta_2 \in GF(2^m),$$

Тогда $\beta_1+\beta_2=1$, $\beta_1\beta_2=\beta$. Поэтому $\beta=\beta_1+\beta_1^2$ и

$$T_2(\beta)=T_2(\beta_1+\beta_1^2)=T_2(\beta_1)+T_2(\beta_1^2)=T_2(\beta_1)+T_2(\beta_1)=0.$$

Установить, какими свойствами следа T_q мы пользовались выше.

11. (Условие приводимости многочлена второй степени.) Если $\beta\in GF(2^m)$ и $T_2(\beta)=0$, то многочлен $X^2+X+\beta$ приводим над полем $GF(2^m)$. В самом деле, рассмотрим отображение $GF(2^m)$ в себя, задаваемое многочленом $f(X)=X^2+X$. При этом отображении каждый элемент δ поля $GF(2^m)$ переходит в элемент $(\delta^2+\delta)$, след которого равен нулю. В $GF(2^m)$ имеется ровно 2^{m-1} элементов, след которых равен 0. Пусть A — множество этих элементов.

Поскольку рассматриваемое отображение $GF(2^m)$ в E сюръективно, то для любого $\beta\in GF(2^m)$, такого, что $T_2(\beta)=0$, справедливо равенство $f(X)=\beta$. Это означает, что в $GF(2^m)$ существуют элементы X (их два), такие, что $X^2+X+\beta=0$, т. е. $X^2+X+\beta$ неприводимым не является.

12. (Преобразование неприводимых многочленов.) Пусть β_0 — элемент $GF(2^m)$, такой, что $T_2(\beta_0)=1$, и пусть $g(X)=X^2+\gamma X+\beta$ — многочлен, неприводимый над $GF(2^m)$. Показать, что при соответствующем выборе β' преобразование $X=\gamma Y+\beta'$ переводит многочлен $g(X)$ в $g(Y)=\gamma^2(Y^2+Y+\beta_0)$.

3. Линейные и циклические коды

3.1. Линейные коды

Линейные коды составляют лишь небольшой подкласс блоковых кодов, однако, поскольку они образуют такую хорошо известную структуру, как линейное пространство, то сразу же после их построения Киясу [1] и Слепяном [2] они стали основным объектом исследования в теории кодирования. Предположим, что число символов на входе канала q является некоторой степенью простого числа p. Входные символы канала представляют собой обозначения для реальных сигналов в канале; для удобства обращения с ними возьмем в качестве входного алфавита V_1 канала конечное поле $GF(q)$ из q элементов. Введенное выше предположение о том, что q является степенью простого числа, обусловлено тем, что, как указывалось в разд. 2.5, число элементов в любом конечном поле есть степень простого числа и, наоборот, для любого q, являющегося степенью простого числа, существует поле из q элементов.

Пусть V_n — векторное пространство размерности n над полем $GF(q)$. Подпространства размерности k пространства V_n называются q-ичными *линейными кодами* длины n с k информационными символами [или (n, k)-кодами]. Если C — линейный код, то подпространства линейного пространства C будем называть (линейными) *подкодами* кода C.

Коды из примеров 1.3 и 1.4 являются линейными. В случае линейных кодов кодовые слова называют также *кодовыми векторами*. При $q = 2$ линейные коды называют *групповыми*.

Как было замечено выше, линейные коды представляют собой лишь подкласс класса блоковых кодов; известны примеры нелинейных кодов, число кодовых слов в которых больше числа кодовых слов в любом линейном коде с теми же параметрами n и d [3—7]. Однако линейные коды обладают более простой структурой, что позволяет выявить более тонкие их свойства и делает их перспективными при построении простых процедур кодирования и декодирования. Кроме того, следует заметить, что коды Препараты [4], являющиеся типичными представителями нелинейных кодов, тесно связаны с циклическими кодами (см. разд. 3.5), являющимися в свою очередь подклассом класса линейных кодов.

Назовем *весом Хэмминга* вектора v (или просто *весом* вектора v) из V_n число ненулевых компонент этого вектора; вес

вектора v будем обозначать через $w(v)$. Расстояние *Хэмминга* $d(v, u)$ между векторами $u, v \in V_n$ определим как вес их разности, а именно положим $d(v, u) = w(v - u)$. Пусть d — минимальное расстояние линейного кода C. Тогда для любого ненулевого вектора $v \in C$

$$w(v) = w(v - 0) \geqslant d,$$

так как нулевой вектор также принадлежит коду C (нулевой вектор обозначен здесь символом 0).

С другой стороны, из определения d следует существование векторов $v, u \in C$, таких, что $d(v, u) = w(v - u) = d$. Но тогда, поскольку $v - u \in C$, *минимальный вес кода* C (который определяется как минимальный из весов ненулевых векторов, входящих в C) в точности равен минимальному расстоянию кода C.

Для задания линейного кода удобно пользоваться определяемой ниже матрицей G. Пусть $g_1, \ldots, g_k$ — некоторый базис (n, k)-кода C, и пусть G — матрица из k строк и n столбцов, i-й строкой которой является базисный вектор g_i. Матрица G называется *порождающей матрицей* кода C. Из свойств базиса непосредственно следует, что любой кодовый вектор может быть представлен в виде линейной комбинации строк $g_1, \ldots, g_k$ матрицы G и, наоборот, что любая линейная комбинация строк $g_1, \ldots, g_k$ матрицы G представляет собой кодовый вектор и, более того, различные линейные комбинации задают различные кодовые векторы.

Поскольку каждый из k коэффициентов линейной комбинации может принимать q значений, общее число кодовых векторов в коде C равно q^k. Наоборот, задав матрицу G размера $k \times n$ и ранга k с элементами из поля $GF(q)$, мы задаем линейный код, для которого G является порождающей матрицей. Об удобстве задания линейных кодов с помощью порождающих матриц свидетельствует, например, тот факт, что двоичный линейный (40, 20)-код содержит более чем 10^6 слов, а задается порождающей матрицей размера всего лишь 20×40.

Для задания линейных кодов используются также матрицы H, которые определяются следующим образом. Пусть C_D — множество всех векторов $u = (u_1, \ldots, u_n) \in V_n$, таких, что для любого $v = (v_1, \ldots, v_n) \in C$

$$uv^T = 0 \quad \left(\text{т. е. } \sum_{i=1}^{n} u_i v_i = 0\right),$$

где индекс T означает транспонирование. Как известно, C_D является подпространством размерности $(n - k)$ пространства V_n. Это подпространство называется кодом, *двойственным* коду C. Порождающую матрицу двойственного кода назовем *прове-*

рочной матрицей (или просто матрицей H) исходного кода C. Матрица H имеет размер $(n-k)\times n$. По определению

$$Hv^T=0 \Leftrightarrow v\in C. \tag{3.1}$$

В частности, если взять в качестве векторов v базисные векторы кода C, то получим

$$HG^T=0,$$

где 0 — нулевая матрица. С помощью матрицы H, например, задавался код в примере 1.4. Поскольку по определению ранг матрицы H равен $(n-k)$, то H содержит невырожденную подматрицу H' размера $(n-k)\times(n-k)$. Путем перестановки столбцов H и соответствующего изменения нумерации компонент кодовых векторов всегда можно добиться того, чтобы матрица H' была образована последними $(n-k)$ столбцами H. Как хорошо известно, с помощью элементарных операций над строками [умножения строк на ненулевые элементы поля $GF(q)$ и сложения одной строки с другой] матрица H всегда может быть приведена к следующему виду:

$$H_0=[P,\ I_{n-k}], \tag{3.2}$$

где P — некоторая матрица размера $(n-k)\times k$, а I_{n-k} — единичная матрица размера $(n-k)\times(n-k)$.

Поскольку H_0 получена посредством элементарных операций над строками, то

$$Hv^T=0 \Leftrightarrow H_0v^T=0, \tag{3.3}$$

т. е. H_0 также является проверочной матрицей кода C. Обозначим через p_{ij} элемент подматрицы P матрицы H_0, стоящий в i-й строке и j-м столбце. Очевидно, что необходимым и достаточным условием принадлежности вектора $v=(v_1, \ldots, v_n)$ коду C является выполнение следующей системы равенств:

$$v_{k+i}=-\sum_{j=1}^{k} p_{ij}v_j, \quad 1\leqslant i\leqslant n-k. \tag{3.4}$$

Таким образом, если компоненты $v_1, \ldots, v_k\in GF(q)$ заданы, то существует ровно один кодовый вектор, первыми k компонентами которого являются $v_1, \ldots, v_k$, и равенства (3.4) задают остальные компоненты этого кодового вектора. Другими словами, мы показали, что любой линейный код является систематическим, и получили при этом равенства (3.4), которые описывают процедуру кодирования. В самом деле, первые k компонент кодового вектора можно считать информационными и использовать для передачи блоков длины k, поступающих от источника, а остальные $(n-k)$ компонент можно вычислять с помощью равенств (3.4).

Утверждение 3.1. *Пусть проверочная матрица линейного* (n, k)*-кода* C *имеет вид* $H_0 = [P, I_{n-k}]$. *Тогда матрица* $G_0 = [I_k - P^T]$ *является порождающей матрицей кода* C. *Справедливо также и обратное утверждение.*

Краткое доказательство. Ранг G_0 равен k. Поскольку $H_0 G_0^T = PI_k - I_{n-k}P = 0$, то каждая строка G_0 является кодовым вектором кода C.

Вернемся к примеру 1.4. Изменяя нумерацию компонент кодовых векторов, можно привести проверочную матрицу рассматриваемого кода к виду (3.2). В данном случае

$$H_0 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Но тогда, как следует из утверждения 3.1, проверочная матрица кода, двойственного (7.4)-коду Хэмминга из примера 1.4, имеет следующий вид:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Пример 3.1. Пусть C — линейный q-ичный (n, k)-код. Линейный q-ичный $(n+1, k)$-код

$$C_{ex} = \left\{ \left(-\sum_{i=1}^{n} v_i, v_1, \ldots, v_n \right) \middle| (v_1, \ldots, v_n) \in C \right\}$$

называется *расширением* кода C. Исходный код C при этом называют *укорочением* расширенного кода C_{ex}. Множество $\left\{ (v_1, \ldots, v_n) \middle| \sum_{i=1}^{n} v_i = 0, (v_1, \ldots, v_n) \in C \right\}$ является $(n, k-1)$-подкодом кода C и называется *четным подкодом* кода C. Если C_{ex} содержит вектор $(1, \ldots, 1)$, то двойственный код C_{exD} кода C_{ex} является расширением двойственного кода $C_D^{(e)}$ четного подкода $C^{(e)}$ кода C.

Доказательство. Пусть $(u_0, u_1, \ldots, u_n) \in C_{exD}$. Поскольку $(1, 1, \ldots, 1) \in C_{ex}$, то

$$u_0 + \sum_{i=1}^{n} u_i = 0.$$

Следовательно, C_{exD} — расширение кода $C_{exD}^{(s)} = \{(u_1, u_2, \ldots, u_n) \mid (u_0, u_1, \ldots, u_n) \in C_{exD}\}$. Число информационных символов

кода $C_{exD}^{(s)}$ совпадает с числом информационных символов кода C_{exD} и равно $n-k+1$. Если $(u_1, \ldots, u_n) \in C_{exD}^{(s)}$ и $(v_1, v_2, \ldots, v_n) \in C^{(e)}$, то

$$\left(\sum_{i=1}^{n} u_i\right)\left(\sum_{i=1}^{n} v_i\right) + \sum_{i=1}^{n} u_i v_i = 0.$$

Так как $\sum_{i=0}^{n} v_i = 0$, то $\sum_{i=1}^{n} u_i v_i = 0$, а это значит, что $(u_1, u_2, \ldots, u_n) \in C_D^{(e)}$. Далее, поскольку число информационных символов кода C_D^e равно $n-k+1$ и совпадает с числом информационных символов кода $C_{exD}^{(e)}$, то $C_D^{(e)} = C_{exD}^{(s)}$.

Приведенные выше положения теории кодирования представляют собой достаточно хорошо известные факты линейной алгебры. Однако с введением понятия веса, которое играет центральную роль в теории кодирования, теория линейных кодов приобретает собственные характерные черты. Приведенная ниже теорема показывает, каким образом минимальный вес кода может быть связан с классическими понятием линейной зависимости.

Обозначим через h_i i-й столбец проверочной матрицы H линейного кода C. Пусть $v = (v_1, \ldots, v_n)$ — кодовый вектор веса w, ненулевые компоненты которого имеют номера $i_1, \ldots, i_w$. Из формулы (3.1) имеем

$$\sum_{j=1}^{w} v_{i_j} h_{i_j} = 0.$$

Следовательно, столбцы H с номерами $i_1, \ldots, i_w$ должны быть линейно зависимыми. Наоборот, предположим, что w столбцов H с номерами $i_1, \ldots, i_w$ линейно зависимы, но любая совокупность меньшего числа из них оказывается линейно независимой. Из определения линейной зависимости следует существование ненулевых элементов $a_j \in GF(q)$, $1 \leqslant j \leqslant w$, таких, что

$$\sum_{j=1}^{w} a_j h_{i_j} = 0. \tag{3.5}$$

Пусть $v = (v_1, \ldots, v_n)$ — вектор, у которого компоненты с номерами i_j, $1 \leqslant j \leqslant w$, равны a_j, а другие равны 0. Тогда в силу равенства (3.5)

$$Hv^T = 0,$$

т. е. v — кодовый вектор веса w.

Таким образом, мы доказали следующую теорему.

Теорема 3.1. *Минимальный вес линейного (n, k)-кода C равен d тогда и только тогда, когда любые $(d-1)$ столбцов проверочной матрицы этого кода линейно независимы, но некоторые d столбцов проверочной матрицы линейно зависимы.*

Прямым следствием этой теоремы является следующее.

Следствие 3.1. $d \leqslant n - k + 1$.

Линейный код, для которого последнее соотношение выполняется со знаком равенства, называется *разделимым кодом с достижимым максимальным расстоянием.* Примерами таких кодов являются описанные ниже коды Рида — Соломона, а также БЧХ-коды из примера 4.5.

Доказанная простая теорема указывает способ прямого построения проверочных матриц H кода с заданными параметрами n, k и d. Она оказывается полезной при построении и исследовании не только БЧХ-кодов и кодов Гоппы, но и кодов Сриваставы [9] и других кодов. Кроме того, на основании этой теоремы выводится граница Варшамова — Гилберта. До сих пор V_1 считалось конечным полем, однако в действительности можно рассмотреть более общий случай, когда V_1 является конечным кольцом. Однако, поскольку в этом более общем случае не удается получить какие-либо полезные результаты, далее мы будем по-прежнему считать, что V_1 — конечное поле.

Утверждение 3.2. *Пусть q — степень простого числа и $n = (q^m - 1)/(q - 1)$, и пусть α — примитивный элемент поля $GF(q^m)$. Представим каждый элемент α^i, $0 \leqslant i < n$, поля $GF(q^m)$ в виде*

$$\alpha^l = \sum_{j=1}^{m} a_{jl}\alpha^{j-1}, \quad a_{jl} \in GF(q).$$

Пусть $A_i^T = (a_{1i}, a_{2i}, \ldots, a_{mi})$. Тогда q-ичный линейный $(n, n-m)$-код, задаваемый проверочной матрицей $H = (A_0 A_1 \ldots A_{n-1})$ размера $m \times n$, является совершенным кодом, исправляющим одиночные ошибки.

Доказательство. Пусть $v = (v_0, v_1, \ldots, v_{n-1})$ — вектор из V_n, такой, что все его компоненты, за исключением v_{i_1} и v_{i_2} $(i_1 < i_2)$, равны нулю. Если предположить, что $Hv^T = 0$, то

$$v_{i_1}A_{i_1} + v_{i_2}A_{i_2} = 0.$$

Но тогда, если один из элементов v_{i_1} или v_{i_2} равен нулю, то и другой элемент также должен быть равен нулю. Если же $v_{i_1}, v_{i_2} \neq 0$, то из определения A_i имеем

$$v_{i_1}\alpha^{i_1} + v_{i_2}\alpha^{i_2} = 0,$$
$$-v_{i_1}v_{i_2}^{-1} = \alpha^{i_2 - i_1},$$

где $0 < i_2 - i_1 < n$ и $-v_{i_1} v_{i_2}^{-1} \in GF(q) - \{0\}$. Возводя обе части последнего равенства в степень $q-1$, получаем

$$1 = \alpha^{(q-1)(i_2-i_1)}.$$

Это приводит к противоречию, поскольку $0 < (q-1)(i_2 - i_1) < q^m - 1$. Следовательно, минимальное расстояние рассматриваемого кода не меньше 3. Более того, поскольку этот код лежит на границе Хэмминга при $t = 1$, он является совершенным. Этот код называется *q-ичным* (*обобщенным*) *кодом Хэмминга*.

3.2. Методы декодирования линейных кодов

Рассмотрим далее методы декодирования линейных кодов. Пусть v — переданный кодовый вектор, а u — соответствующий ему принятый вектор (получающийся в результате разбиения последовательности символов на выходе демодулятора на блоки длины n в соответствии с разбиением на блоки передаваемой последовательности). Вектор $e = u - v$ называется *вектором ошибок*, а вектор $uH^T \in V_{n-k}$ — *синдромом*, соответствующим вектору $u \in V_n$. Как следует из формулы (3.1), $vH^T = 0$. Поэтому

$$uH^T = (u - v) H^T = eH^T. \tag{3.6}$$

Векторы x, $y \in V_n$ называются сравнимыми по модулю C, если $(x - y) \in C$ [12]; сравнимость векторов обозначается следующим образом: $x \equiv y \pmod{C}$. Из определения сравнимости векторов непосредственно следует, что: 1) $x \equiv x \pmod{C}$; 2) если $x \equiv y \pmod{C}$, то $y \equiv x \pmod{C}$ и 3) если $x \equiv y$, $y \equiv z \pmod{C}$, то $x \equiv z \pmod{C}$. В этом случае, как хорошо известно, множество V_n можно разбить на непересекающиеся классы таким образом, что любые два вектора из одного класса будут сравнимы по модулю C, но любые два вектора из разных классов не будут сравнимыми по модулю C. Эти классы называют смежными классами (см. утверждение 2.5 и раздел, посвященный нормальным делителям). Смежный класс, содержащий вектор x, будем обозначать через (x). Заметим, что смежный класс (0) совпадает с кодом C. В общем случае

$$(x) = \{y \mid y = x + v,\ v \in C\}.$$

Это соотношение показывает, что каждый смежный класс состоит из q^k векторов. Число различных смежных классов равно q^{n-k}. Так как в силу (3.1)

$$x \equiv y \pmod{C} \Leftrightarrow x - y \in C \Leftrightarrow xH^T = yH^T,$$

то необходимым и достаточным условием того, что векторы x и y принадлежат одному смежному классу, является равенство

соответствующих им синдромов xH^T и yH^T. Другими словами, синдром s вектора x однозначно определяет смежный класс, которому принадлежит x. Обозначим смежный класс, соответствующий синдрому s, через $C(s)$.

Предположим, что все кодовые векторы передаются с равными вероятностями и вероятность получить при приеме вектор $v+e$, где v — переданный кодовый вектор, зависит только от вектора ошибок e. Пусть $e^{(1)}$ и $e^{(2)}$ — векторы ошибок, принадлежащие одному смежному классу $(e^{(1)})=(e^{(2)})$. Предположим, что передавался кодовый вектор v, а был принят вектор $v+e^{(1)}$. Так как $e^{(1)}-e^{(2)}\in C$, то в действительности при приеме есть все основания предположить, что передавался кодовый вектор $u=v+e^{(1)}-e^{(2)}$, вектор ошибок был равен $e^{(2)}$, а принят вектор $u+e^{(2)}(=v+e^{(1)})$. При этом должно быть принято то решение, которое имеет наибольшую вероятность оказаться правильным. Следовательно, из всех векторов ошибок, принадлежащих одному смежному классу, должен быть выбран тот, который имеет наибольшую вероятность. Этот вектор называется представителем или лидером своего смежного класса. Как указывалось выше, в ДСК или канале, являющемся обобщением ДСК на q-ичный случай, лидером смежного класса является вектор минимального веса.

Для декодирования линейного кода можно построить таблицу декодирования, которая каждому вектору $s\in V_{n-k}$ сопоставляет лидер смежного класса $C(s)$. При этом декодирование линейного кода осуществляется следующим образом:

1) По принятому вектору u вычисляется синдром $s=uH^T$.

2) С помощью таблицы декодирования находится вектор ошибок e, соответствующий данному s; кодовое слово $u-e$ посылается на выход декодера.

Задача 3.3. Описать таблицу декодирования в случае, если часть ошибок декодируется (исправляется), а другая часть обнаруживается.

Пример 3.2. Рассмотрим двоичный линейный $(6,3)$-код, задаваемый проверочной матрицей

$$H=\begin{bmatrix}1&0&1&1&0&0\\0&1&1&0&1&0\\1&1&1&0&0&1\end{bmatrix}.$$

Этот код получается из $(7,3)$-кода Хэмминга, описанного в примере 1.4 из разд. 1.1, удалением третьей компоненты во всех кодовых словах и изменением нумерации остальных компонент. В первой строке табл. 3.1 перечислены все кодовые векторы рассматриваемого кода. Поскольку таблица имеет большие раз-

Таблица 3.1

Смежные классы кода из примера 3.2

Синдром	Лидеры смежных классов							
000	(00)	(17)	(23)	(34)	(45)	(52)	(66)	(71)
001	(01)	(16)	(22)	(35)	(44)	(53)	(67)	(70)
010	(02)	(15)	(21)	(36)	(47)	(50)	(64)	(73)
100	(04)	(13)	(27)	(30)	(41)	(56)	(62)	(75)
111	(10)	(07)	(33)	(24)	(55)	(42)	(76)	(61)
011	(20)	(37)	(03)	(14)	(65)	(72)	(46)	(51)
101	(40)	(57)	(63)	(74)	(05)	(12)	(26)	(31)
110	(06)	(11)	(25)	(32)	(43)	(54)	(60)	(77)

меры, то двоичные векторы записаны в восьмеричном представлении; например, вектору (101110) соответствует запись (56). В каждой из других строк этой таблицы перечислены векторы одного из смежных классов. Так как этот код исправляет все одиночные ошибки (см. пример 1.4), то все векторы веса 1 можно взять в качестве лидеров смежных классов. В последнем смежном классе вектор (000110) является вектором минимального веса, так что его можно выбрать в качестве лидера этого смежного класса. Таким образом, рассматриваемый код исправляет все одиночные ошибки и одну ошибку веса 2. При использовании этого кода в ДСК для исправления ошибок вероятность правильного декодирования равна

$$p_0^6 + 6p_0^5q_0 + p_0^4q_0^2$$

(предполагается, что все кодовые векторы посылаются с равными вероятностями).

Так как сам код C как смежный класс соответствует отсутствию ошибок, то в качестве лидеров $q^{n-k} - 1$ остальных смежных классов желательно иметь векторы, которые как векторы ошибок имеют наибольшие вероятности появления. Если в качестве лидеров q^{n-k} смежных классов удается взять все векторы веса t и меньше и только их, то код C является совершенным, так как

$$\sum_{i=1}^{t} \binom{n}{i} (q-1)^i = q^{n-k} - 1.$$

Как указывалось в разд. 1.4.1, совершенных кодов существует не очень много. Более широким является класс так называемых квазисовершенных кодов. *Квазисовершенными* называются такие линейные коды, в качестве лидеров q^{n-k} смежных классов которых можно взять все векторы веса t и менее и несколько

векторов веса $t+1$. Если канал является ДСК или обобщением ДСК на q-ичный случай, то в классе линейных кодов с одинаковыми параметрами квазисовершенные коды имеют минимальную вероятность ошибки декодирования. Квазисовершенные коды, так же как и совершенные коды, могут не существовать при некоторых значениях n и d (например, двоичные квазисовершенные коды с параметрами $n=15$, $d=5$ и $n=17$, $d=5$ существуют, а двоичный квазисовершенный код с параметрами $n=16$, $d=5$ не существует [10]), однако квазисовершенных кодов существует гораздо больше, чем совершенных. Например, квазисовершенными являются код из примера 3.2 и все двоичные БЧХ-коды, исправляющие ошибки веса 2 [11]. Кроме того, при $d=5$ и $q=2$ всегда существует по крайней мере один квазисовершенный код с любым наперед заданным числом проверочных символов (утверждение 3.9).

Число строк описанной выше таблицы декодирования линейного кода равно q^{n-k}. По сравнению с числом q^n строк таблицы декодирования произвольного блокового кода число q^{n-k} строк в описанной таблице значительно меньше, однако, например при $q=2$ и $n-k=30$, таблица декодирования линейного кода содержит более чем 10^9 строк, что, конечно, остается практически неприемлемым. Вполне естественно поэтому перейти к исследованию более полезных кодов, для которых существуют практически приемлемые алгоритмы нахождения вектора ошибок по синдрому. Ниже в разд. 3.5 рассматривается один из таких классов кодов.

Утверждение 3.4. *Если в линейном двоичном (n, k)-коде существуют кодовые векторы нечетного веса, то их число равно 2^{k-1}. Более того, множество всех кодовых векторов четного веса образует подкод C_e размерности $k-1$.*

Краткое доказательство. Сначала следует доказать вторую часть утверждения. Далее, если $v_{\text{нечетн}}$ — один из кодовых векторов нечетного веса, то множество $\{v_{\text{нечетн}}+u \mid u \in C_e\}$ является смежным классом по подгруппе C_e. Этот смежный класс состоит только из кодовых векторов нечетного веса, а их число равно 2^{k-1}.

Утверждение 3.5. *Если хотя бы один из кодовых векторов q-ичного линейного (n, k)-кода имеет ненулевую i-ю компоненту, то число кодовых векторов, i-я компонента которых равна элементу α поля $GF(q)$, составляет q^{k-1}.*

Краткое доказательство. Множество всех кодовых векторов, i-я компонента которых равна нулю, является подпространством размерности $(k-1)$. Для доказательства утверждения следует

рассмотреть порождаемое этим подпространством разложение рассматриваемого кода на смежные классы. Множество кодовых векторов, имеющих одну и ту же i-ю компоненту, является одним из смежных классов этого разложения.

Утверждение 3.6 [8]. *Если для каждого целого* i, $1 \leqslant i \leqslant n$, *в коде существует хотя бы один кодовый вектор,* i*-я компонента которого не равна нулю, то сумма весов всех кодовых векторов* q*-ичного линейного* (n, k)*-кода равна* $n(q-1)q^{k-1}$.

Идея доказательства. Следует воспользоваться тем, что, согласно утверждению 3.5, число кодовых векторов, i-я компонента которых не равна нулю, составляет $(q-1)q^{k-1}$.

Утверждение 3.7. *Пусть* C *представляет собой* q*-ичный линейный* (n, k)*-код.* Тогда:

1) *если код* C *обнаруживает пачки ошибок длины* b *и менее, то*

$$n - k \geqslant b;$$

2) *если код* C *исправляет пачки ошибок длины* b *и менее, то*

$$n - k \geqslant 2b.$$

Эти соотношения известны как нижние границы Рейджера [12].

Краткое доказательство. Пусть V_n^i — множество векторов из V_n, все компоненты которых с $(i+1)$-й по n-ю равны нулю. Рассмотрим случай 1. Предположим, что два различных вектора из V_n^b принадлежат одному и тому же смежному классу по подгруппе C. Тогда существует ненулевой вектор $v \in V_n^b$, принадлежащий коду C, т. е. код C не обнаруживает по крайней мере одну пачку ошибок длины b или менее, какой является вектор v. Следовательно, число смежных классов не может быть меньше, чем q^b. Отсюда следует, что $n - k \geqslant b$. Рассмотрим случай 2. Аналогично можно доказать, что никакие два различных вектора из V_n^{2b} не должны входить в один смежный класс. Следовательно, число смежных классов не может быть меньше, чем q^{2b}, т. е. $n - k \geqslant 2b$.

Утверждение 3.8 [13]. *Пусть* C *— двоичный линейный* (n, k)*-код с минимальным расстоянием* d *и порождающей матрицей* G. *Предположим, что среди лидеров смежных классов (в качестве которых берутся векторы минимального веса) существуют векторы веса* $d-1$ *или более; пусть* u *— один из таких лидеров. Тогда минимальное расстояние двоичного линейного* $(n+1,$

$k+1)$-*кода* C', *задаваемого порождающей матрицей* $G' = \left[\begin{array}{cc} & 0 \\ G & \vdots \\ & 0 \\ u & 1 \end{array}\right]$, *равно* d.

Краткое доказательство. Пусть $v = (v_1, v_2, \ldots, v_n, v_{n+1})$ — кодовый вектор кода C'. Если $v_{n+1} = 0$, то вес v не меньше чем d, так как $(v_1, \ldots, v_n) \in C$. Если $v_{n+1} = 1$, то $(v_1, v_2, \ldots, v_n) = (u_1, u_2, \ldots, u_n) + (v_1', v_2', \ldots, v_n')$, где $(v_1', v_2', \ldots, v_n') \in C$. Это равенство показывает, что вектор $(v_1, v_2, \ldots, v_n)$ принадлежит тому же смежному классу, что и вектор $(u_1, u_2, \ldots, u_n)$, а следовательно, вес $(v_1, v_2, \ldots, v_n)$ не меньше чем $d-1$.

Утверждение 3.9 [13]. *В утверждении* 3.8 *положим* $d = 5$. *Если код* C *не является квазисовершенным, то условия утверждения* 3.8 *выполняются, так что существует код* C' *с* $d = 5$. *Если в свою очередь код* C' *не является квазисовершенным, то по* C' *описанным в утверждении* 3.8 *способом можно построить новый код и т. д. Эта процедура всегда приводит к квазисовершенному коду.*

Краткое доказательство. Заметим, что каждый вновь построенный код имеет то же число проверочных символов, что и исходный код. Граница Хэмминга в этом частном случае имеет следующий вид:

$$\log_2\left(1 + n + \frac{n(n+1)}{2}\right) \leqslant r.$$

Из этого неравенства видно, что описанная выше процедура построения последовательности кодов должна закончиться через конечное число шагов.

Утверждение 3.10 [8]. *Двоичные квазисовершенные коды длины* n, *исправляющие одиночные ошибки, существуют для всех* n, *заключенных в пределах* $2^{m/2+1} - 2 \leqslant n < 2^m$, *где* m — *произвольное четное число, и* $2^{(m+1)/2} + 2^{(m-1)/2} - 2 \leqslant n \leqslant 2^m$, *где* m — *произвольное нечетное число.*

Краткое доказательство. Квазисовершенными кодами длины $n = 2^{m/2+1} - 2$ при четном m и $n = 2^{(m+1)/2} + 2^{(m-1)/2} - 2$ при нечетном m являются коды, столбцами проверочных матриц размера $m \times n$ которых являются: 1) все вектор-столбцы, первые $[m/2]$ компонент которых равны нулю, и 2) все вектор-столбцы, последние $m - [m/2]$ компонент которых равны нулю (нулевые векторы в число столбцов проверочной матрицы не включаются). Поскольку произвольный вектор-столбец длины m может быть представлен в виде суммы вектор-столбца первого

типа и вектор-столбца второго типа, то для любого двоичного вектора v длины n найдется вектор u длины n веса 2 или менее, такой, что их сумма $v+u$ будет кодовым словом. Следовательно, все лидеры смежных классов рассматриваемого кода имеют вес 2 или менее.

3.3. Нижняя граница Варшамова — Гилберта

В разд. 1.4 выше были получены верхние границы для произвольных блоковых кодов. Здесь мы получим нижнюю границу Варшамова — Гилберта, которая гарантирует существование линейного кода с заданным числом проверочных символов $r = n-k$ и заданным минимальным расстоянием d, число информационных символов которого больше определенной величины (вначале Гилберт получил эту границу для блоковых кодов [14]; Варшамов улучшил результат Гилберта, получив аналогичную границу для линейных кодов [15, 16]).

Теорема 3.2. *Пусть q — степень простого числа, а r и d — некоторые целые положительные числа. Тогда существует q-ичный линейный код длины n с r проверочными символами и кодовым расстоянием, не меньшим d, параметры которого удовлетворяют неравенству*

$$q^r \leqslant \sum_{i=0}^{d-2} \binom{n}{i} (q-1)^i. \tag{3.7}$$

Используя функцию φ из неравенства Чернова, неравенство (3.7) *можно заменить следующим*:

$$\frac{r}{n} \leqslant \varphi\left(\frac{d-2}{n}\right). \tag{3.8}$$

В частности, при $q=2$ последнее неравенство имеет вид

$$\frac{r}{n} \leqslant H\left(\frac{d-2}{n}\right). \tag{3.8'}$$

Доказательство. Как следует из теоремы 3.1, построение матрицы H, любые $d-1$ столбцов которой линейно независимы, эквивалентно построению кода с минимальным расстоянием, не меньшим d. Построим матрицу H, удовлетворяющую указанному выше свойству, следующим образом. Вначале произвольным образом выберем в множестве $V_r' = V_r -$ {нулевой вектор} 1-й столбец h_1 матрицы H.

В качестве 2-го столбца h_2 матрицы H возьмем произвольный вектор из V_r', не являющийся произведением ah_1, $a \in GF(q)$. Если в V_r' существует хотя бы один вектор, не являющийся линейной комбинацией h_1 и h_2, то выберем один из этих векто-

ров (произвольный) в качестве h_3. Предположим, что таким образом мы выбрали j векторов h_i, $0 \leqslant i \leqslant j$; по построению h_i не является линейной комбинацией никаких $d-2$ или менее столбцов из $h_1, \ldots, h_{i-1}$. Поскольку i $(0 \leqslant i \leqslant d-2)$ векторов из $h_1, \ldots, h_j$ можно выбрать $\binom{j}{i}$ способами, а число способов выбора i ненулевых коэффициентов линейной комбинации равно $(q-1)^i$, то число векторов, являющихся линейными комбинациями $d-2$ или менее столбцов из $h_1, \ldots, h_j$ не больше

$$\sum_{i=1}^{d-2} \binom{j}{i} (q-1)^i.$$

Если это число меньше общего числа векторов в V_r', то в V_r' существует вектор, не совпадающий ни с одной из указанных выше линейных комбинаций. Этот вектор можно выбрать в качестве h_{j+1}. К тому моменту, когда новый вектор h_{n+1} выбрать уже нельзя, общее число n выбранных ранее векторов h_i удовлетворяет неравенству (n является длиной кода)

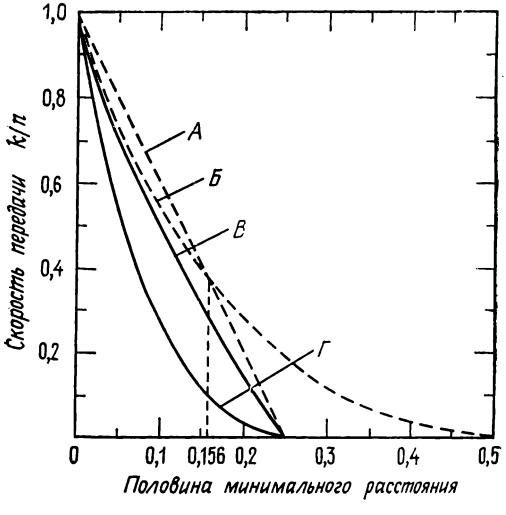
$$\sum_{i=1}^{d-2} \binom{n}{i} (q-1)^i \geqslant q^r - 1.$$

Из этого неравенства и границы Чернова получаются неравенства (3.8) и (3.8′).

На фиг. 3.1 показано поведение верхних границ Плоткина, Хэмминга и Элайса, а также нижней границы Варшамова — Гилберта при достаточно больших n (здесь по вертикальной оси откладывается скорость кода $R = k/n$, а по горизонтальной — отношение минимального расстояния $d_{\text{мин}}$ к $2n$ [8].

Вывод неравенства (3.7), как мы видели выше, допускает любой способ выбора векторов $h_1, \ldots, h_i, \ldots$, удовлетворяющих условиям теоремы 3.1. Поэтому если задан линейный (n, k)-код с минимальным весом d, параметры n, k и d которого удовлетворяют неравенству (3.7), то к проверочной матрице этого кода всегда можно добавить еще один столбец, не нарушая при этом условий теоремы 3.1. Другими словами, сохраняя постоянным и число проверочных символов, и минимальный вес и увеличивая число информационных символов, можно расширить исходный код. Как показывают несложные расчеты, минимальный вес почти всех линейных кодов близок к величине, гарантируемой неравенством (3.7) [8]. Из неравенства (3.8′) следует, что при фиксированном отношении d/n $(<1/2)$ и $n \to \infty$ существуют линейные коды, для которых отношение k/n больше $1 - H(d/n)$. Здесь следует заметить, что для БЧХ-кодов, обладающих хорошими свойствами с точки зрения кодирования и декодирования, при фиксированном d/n и $n \to \infty$ отношение

k/n стремится к нулю (см. разд. 4.1). За исключением кодов Юстесена [14], все известные к настоящему времени алгебраические коды обладают этим недостатком. Однако и для кодов Юстесена отношение k/n значительно меньше величины, гарантируемой границей Варшамова — Гилберта. Построение лежащих на границе Варшамова — Гилберта (или близко к ней) кодов, способ построения которых можно было бы задать явно (конструктивно) и которые обладали бы хорошими процедурами кодирования и декодирования, является одной из нерешенных проблем теории кодирования [1]). Так как параметры двоичных



Фиг. 3.1.

А — верхняя граница Плоткина; *Б* — верхняя граница Хэмминга; *В* — верхняя граница Элайса; *Г* — нижняя граница Варшамова — Гилберта.

БЧХ-кодов с длиной порядка нескольких тысяч являются не столь уж плохими [8] по сравнению с параметрами, задаваемыми (3.8), то исследование асимптотических свойств кодов при больших длинах представляет главным образом теоретический интерес.

3.4. Распределение весов

Рассмотрим q-ичный линейный (n, k)-код, содержащий A_i кодовых векторов веса i. Совокупность чисел $A_0, A_1, \ldots, A_n$ на-

[1]) Эти вопросы более подробно освещены в книге Э. Л. Блоха и В. В. Зяблова [39] и в статье Л. А. Бассалыго [40]. — *Прим. ред.*

зывается *распределением весов*, а многочлен

$$A(z) = \sum_{i=0}^{n} A_i z^i$$

— *нумератором весов* кода C.

Предположим, что код C является линейным, имеет минимальное расстояние d и используется в двоичном симметричном канале (ДСК) с вероятностью ошибки в символе p_0. Выразим вероятность ошибочного декодирования P_{de} и вероятность обнаружения ошибки (отказа от декодирования) P_{df} через нумератор весов кода C [9, 17]. Сначала рассмотрим следующую процедуру декодирования: принятые кодовые векторы декодируются в самих себя, а при получении любого другого вектора, не являющегося кодовым, констатируется лишь факт обнаружения ошибки. В этом случае вероятность ошибочного декодирования совпадает с вероятностью того, что вектор ошибки является одним из ненулевых кодовых векторов, которая в свою очередь равна

$$\sum_{i=1}^{n} A_i p_0^i (1-p_0)^{n-i} = (1-p_0)^n A[p_0/(1-p_0)] - (1-p_0)^n.$$

Поскольку вероятность обнаружения ошибки (отказа от декодирования) P_{df} может быть получена вычитанием P_{de} из вероятности того, что вектор ошибок является ненулевым, то

$$P_{df} = 1 - (1-p_0)^n A[p_0/(1-p_0)].$$

Далее предположим, что $d \leqslant 2t+1$, и рассмотрим следующую процедуру декодирования. Если принятый вектор находится на расстоянии t или менее от одного из кодовых слов (такое кодовое слово, если оно существует, может быть только одно), то он декодируется в это кодовое слово. В остальных случаях ошибки обнаруживаются, но декодирование не производится. Так как в этом случае вероятность правильного декодирования P_{dc} равна вероятности того, что в канале возникнет не более чем t ошибок, то

$$P_{dc} = \sum_{i=0}^{t} \binom{n}{i} p_0^i (1-p_0)^{n-i}. \tag{3.9}$$

Вероятность ошибочного декодирования равна вероятности того, что вектор ошибок в канале принадлежит смежному классу с минимальным весом t или менее (или, другими словами, смежному классу, лидер которого имеет вес t или менее), но не является лидером этого смежного класса. Рассмотрим совокупность векторов $u \in V_n$ веса w, находящихся на расстоянии j от вектора v веса i. Число таких векторов обозначим через $N_{j,w}^{(i)}$.

Очевидно, что $N^{(i)}_{j,w}$ не зависит от того, какие компоненты v равны 1. Для простоты в качестве v возьмем вектор, у которого все компоненты с 1-й по i-ю равны 1, а остальные равны 0. Заметим, что если l — число равных единице компонент вектора u с номерами от 1 до i, то $2l = w + i - j$. Поэтому [1])

$$N^{(i)}_{j,w} = \binom{i}{(w+i-j)/2}\binom{n-i}{(w-i+j)/2}.$$

Введем многочлен $N^{(i)}(x, y) = \sum_j \sum_w N^{(i)}_{j,w} x^j y^w$. Легко проверить, что

$$N^{(i)}(x, y) = \sum_{\nu=0}^{i}\sum_{\mu=0}^{n-i}\binom{i}{\nu}\binom{n-i}{\mu}x^{i-\nu+\mu}y^{\nu+\mu} = (x+y)^i(1+xy)^{n-i}. \tag{3.10}$$

Так как $2t + 1 \leqslant d$, то при $j \leqslant t$ ни один вектор не может находиться на расстоянии j или меньше сразу от двух кодовых векторов. Поэтому общее число векторов веса w, находящихся на расстоянии j ($j \leqslant t$) от кодовых векторов, равно

$$\sum_{i=0}^{n} A_i N^{(i)}_{j,w}, \quad j \leqslant t.$$

Отсюда в свою очередь следует, что общее число векторов веса w, принадлежащих смежным классам с минимальным весом t и менее, равно

$$\sum_{j=0}^{t}\sum_{i=0}^{n} A_i N^{(i)}_{j,w}, \quad j \leqslant t.$$

Но тогда вероятность того, что вектор ошибок принадлежит смежному классу с минимальным весом t и менее, т. е. вероятность того, что принятое слово будет неправильно декодировано, равна

$$\sum_{w=0}^{n}\sum_{j=0}^{t}\sum_{i=j}^{n} A_i N^{(i)}_{j,w} p_0^w (1-p_0)^{n-w}.$$

Для удобства расчета последней вероятности часто используется многочлен

$$P_t(x) = \sum_{j=0}^{t}\sum_{w=0}^{n}\sum_{i=0}^{n} A_i N^{(i)}_{j,w} p_0^w (1-p_0)^{n-w} x^j.$$

---

[1]) В случае когда $n < m$ или $m < 0$, полагаем $\binom{n}{m} = 0$.

Из определения $P_t(x)$ непосредственно следует, что

$$P_{dc} + P_{de} = P_t(1). \tag{3.11}$$

Многочлен $P_t(x)$ является суммой первых t членов вида $A_j x^j$ (степени $j < t$) следующего многочлена:

$$P_n(x) = \sum_{j=0}^{n} \sum_{w=0}^{n} \sum_{i=0}^{n} A_i N_{j,w}^{(i)} p_0^w (1-p_0)^{n-w} x^j.$$

Используя формулу (3.10), многочлен $P_n(x)$ путем несложных преобразований можно привести к виду

$$P_n(x) = \sum_{i=0}^{n} A_i (x - xp_0 + p_0)^i (1 - p_0 + xp_0)^{n-i} =$$

$$= (1 - p_0 + xp_0)^n A\left(\frac{x - xp_0 + p_0}{1 - p_0 + xp_0}\right).$$

Так как $P_{df} = 1 - P_{dc} - P_{de}$, то, зная $A(x)$, с помощью формул (3.9) и (3.11) можно найти P_{dc}, P_{de} и P_{df}. Это показывает, что нумератор весов является одной из важнейших характеристик кода. Однако нумераторы весов не дают полной информации о распределении весов лидеров смежных классов веса $d/2$ и более. Более того, существуют коды с одним и тем же нумератором весов, имеющие, однако, различные распределения весов лидеров смежных классов [9].

Для некоторых классов кодов, например разделимых кодов с максимально достижимым расстоянием, БЧХ-кодов, исправляющих t-кратные ошибки (только при $t \leqslant 3$), и ряда других, известны общие формулы для распределения весов [9, 18—26]. Однако определение распределения весов произвольного кода является непростой задачей. Конечно, для кодов с небольшим числом информационных символов, используя вычислительную машину, с помощью порождающей матрицы можно механически породить все кодовые слова и найти таким образом распределение весов. При небольших значениях числа проверочных символов $(n-k)$ для нахождения распределения весов можно воспользоваться следующим тождеством Мак-Вильямс [17].

Теорема 3.3. *Пусть C представляет собой q-ичный линейный (n, k)-код, C_D — код, двойственный к C, а $A(z)$ и $B(z)$ — соответственно нумераторы весов кодов C и C_D. Тогда*

$$A(z) = q^{k-n}(1 + (q-1)z)^n B\left(\frac{1-z}{1+(q-1)z}\right).$$

Доказательство. Множество векторов, у которых фиксированные j компонент равны 0, очевидно, является подпространством. Выбирая различными способами положение этих j компонент,

можно построить $\binom{n}{j}$ таких подпространств $S^{(m)}$, $1 \leqslant m \leqslant \binom{n}{j}$. Обозначим через $S_D^{(m)}$ подпространство, двойственное к $S^{(m)}$; $S_D^{(m)}$ состоит из векторов, фиксированные $n-j$ компонент которых (дополнительные к j нулевым фиксированным компонентам векторов из $S^{(m)}$) равны 0. Положим $|v \cap S^{(m)}| = 1$, если вектор v принадлежит $S^{(m)}$, и $|v \cap S^{(m)}| = 0$ в противном случае. Если вектор v имеет вес i и $n-i \geqslant j$, то для подпространств $S^{(m)}$, которые определяются выбором j компонент из $n-i$ нулевых компонент вектора v, имеем $|v \cap S^{(m)}| = 1$; $|v \cap S^{(m)}| = 0$ для всех остальных подпространств $S^{(m)}$. Следовательно,

$$\sum_{m=1}^{\binom{n}{j}} |v \cap S^{(m)}| = \binom{n-i}{j}.$$

Но тогда

$$\sum_{m=1}^{\binom{n}{j}} |C \cap S^{(m)}| = \sum_{v \in C} \sum_{m=1}^{\binom{n}{j}} |v \cap S^{(m)}| = \sum_i A_i \binom{n-i}{j}. \quad (3.12)$$

Аналогично получаем

$$\sum_{m=1}^{\binom{n}{j}} |C_D \cap S_D^{(m)}| = \sum_{v \in C_D} \sum_{m-1}^{\binom{n}{j}} |v \cap S_D^{(m)}| = \sum_i B_i \binom{n-i}{n-j}. \quad (3.13)$$

С другой стороны, из утверждения 2.35 следует, что

$$|C \cap S^{(m)}| = q^{k-j} |C_D \cap S_D^{(m)}|. \quad (3.14)$$

Из формул (3.12) и (3.13) имеем

$$\sum_i A_i \binom{n-i}{j} = q^{k-j} \sum_i B_i \binom{n-i}{n-j}. \quad (3.15)$$

Заметим, что j в последнем равенстве является произвольным целым числом из интервала $[0, n]$. Умножим обе части (3.15) на x^j и просуммируем по j. При этом в левой части получим следующую сумму:

$$\sum_j \sum_i A_i \binom{n-i}{j} x^j = \sum_i A_i (1+x)^{n-i}.$$

В то же время сумма в правой части будет равна

$$\sum_j q^{k-j} \sum_i B_i \binom{n-i}{n-j} x^j = q^k \left(\frac{x}{q}\right)^n \sum_i B_i \left(1 + \frac{q}{x}\right)^{n-i}.$$

Разделив обе части получающегося в результате равенства на $(1+x)^n$, получим

$$\sum_i A_i \left(\frac{1}{1+x}\right)^i = q^{k-n}\left(\frac{q+x}{1+x}\right)^n \sum_i B_i \left(\frac{x}{q+x}\right)^i.$$

Замена $z = 1/(1+x)$, $x = (1-z)/z$ переводит последнее равенство в тождество Мак-Вильямс.

Если часть параметров A_i и B_j известна, то для нахождения оставшихся можно воспользоваться тождествами Плесс [8, 9, 27], представляющими собой модифицированные тождества Мак-Вильямс.

Понятие двойственного кода может быть обобщено на случай нелинейных кодов. При этом также удается получить аналог тождества Мак-Вильямс [28].

Утверждение 3.11 [8]. *Пусть G — матрица размера $m \times (2^m - 1)$ с элементами из поля $GF(2)$, все столбцы которой являются различными и ненулевыми. Тогда все ненулевые кодовые векторы двоичного линейного $(2^m - 1, m)$-кода, для которого матрица G является порождающей, имеют вес $2^m - 1$.*

Краткое доказательство. Пусть $v_1, v_2, \ldots, v_m$ — строки G. Рассмотрим линейную комбинацию

$$v_{i_1} + v_{i_2} + \ldots + v_{ij} (i_1 < i_2 < \ldots < i_j,\ 1 \leqslant j \leqslant m).$$

Число столбцов матрицы G, компоненты которых, расположенные в строках с номерами $i_1, i_2, \ldots i_j$, равны заданным двоичным символам $a_1, a_2, \ldots, a_j$, составляет $2^{m-j} - 1$, если $a_1 = a_2 = \ldots = a_j = 0$, и 2^{m-j} во всех остальных случаях. Общее число наборов значений $a_1, a_2, \ldots, a_j$, таких, что $a_1 + a_2 + \ldots + a_j = 1$, равно 2^{j-1}. Отсюда следует, что вес вектора $v_{i_1} + v_{i_2} + \ldots + v_{i_j}$ всегда равен 2^{m-1}.

Задача 3.12. Обобщить предыдущее утверждение на случай произвольного q.

Утверждение 3.13. *Нумератором весов кода, двойственного к коду из утверждения* 3.11, *является многочлен*

$$2^{-m}\{(1+z)^{2^m-1} + (2^m - 1)(1+z)^{2^{m-1}-1}(1-z)^{2^{m-1}}\}.$$

(Для доказательства этого утверждения следует воспользоваться утверждением 3.11 и тождеством Мак-Вильямс.)

3.5. Циклические коды (I)

Циклические коды являются линейными кодами, которые не изменяют своих свойств при циклической перестановке компонент кодовых слоев. Благодаря этому свойству они обладают сравнительно простыми процедурами кодирования и декодирования и, кроме того, имеют удобные алгебраические свойства. Все совершенные линейные коды, коды Рида — Маллера (см. разд. 4.5) и другие коды, построенные еще до того, как Прейндж ввел понятие циклического кода и описал свойства последнего, являются кодами, которые после перестановки компонент или незначительной модификации оказываются циклическими. Все важнейшие блоковые коды, открытые позже, также оказались либо циклическими кодами, либо кодами, получающимися из циклических, либо тесно связанными с последними.

Пусть C представляет собой q-ичный линейный (n, k)-код. Если вместе с каждым кодовым вектором $v = (v_0, v_1, \ldots, v_{n-1})$ (в циклических кодах удобно нумеровать компоненты, начиная с нуля) коду C принадлежит также вектор $Pv = (v_{n-1}, v_0, v_1, \ldots, v_{n-2})$, то код C называется *циклическим*. При изучении циклических кодов векторы $(v_0, v_1, \ldots, v_{n-1})$ удобно представлять в виде многочленов $v(X) = v_0 + v_1X + v_2X^2 + \ldots + v_{n-1}X^{n-1}$ от неизвестного X с коэффициентами из поля $GF(q)$. При этом вектору $Pv = (v_{n-1}, v_0, \ldots, v_{n-2})$ соответствует многочлен

$$Xv(X) - v_{n-1}(X^n - 1).$$

Первая и последняя компоненты кодовых слов циклических кодов рассматриваются как соседние компоненты. Поэтому при представлении кодовых слов циклических кодов в виде многочленов степени X^n и $X^0 = 1$ неизвестного X должны рассматриваться как тождественные. Другими словами, если два многочлена $a(X)$ и $b(X)$ при делении на $X^n - 1$ имеют равные остатки, т. е. если $a(X)$ и $b(X)$ принадлежат одному классу вычетов в кольце многочленов по модулю $X^n - 1$, то они рассматриваются как тождественные. Различные многочлены степени $n - 1$ и менее принадлежат различным классам вычетов по модулю $X^n - 1$. Класс вычетов по модулю $X^n - 1$, содержащий многочлен $f(X)$, будем обозначать через $(f(X))$. Как было описано в разд. 2.4, можно определить операции сложения, вычитания и умножения классов вычетов так, что для этих операций будут выполняться обычные правила арифметики. Множество всех классов вычетов по модулю $X^n - 1$ далее будем обозначать через R_n.

Резюмируя вышеизложенное, заметим, что сопоставление кодовым векторам $(v_0, v_1, \ldots, v_{n-1})$ классов вычетов

$$(v_0 + v_1X + \ldots + v_{n-1}X^{n-1}) \in R_n$$

для циклических кодов является важным по существу. Действительно, это отображение является взаимно однозначным. Далее, вектору Pv соответствует класс вычетов $(Xv(X)) = (X)(v(X))$. И наконец, если при представлении кодовых векторов в виде последовательностей для нахождения линейной комбинации необходимо рассматривать циклические перестановки (соответствующие операциям над матрицами), то при представлении кодовых векторов в виде элементов R_n нахождение линейной комбинации осуществляется только с помощью операций сложения, вычитания и умножения многочленов.

Пусть I — подмножество R_n, соответствующее коду C, а именно

$$I = \{(v(X)) \in R_n \mid v \in C\}.$$

Так как C является линейным подпространством, то для произвольных $(v(X))$, $(u(X)) \in I$ и $a \in GF(q)$

$$(v(X) - u(X)) = (v(X)) + (u(X)) \in I, \tag{3.16}$$

$$(au(X)) = (a)(u(X)) \in I. \tag{3.17}$$

Кроме того, как указывалось выше,

$$(X)(v(X)) \in I. \tag{3.18}$$

Из формул (3.16)—(3.18) следует, что для произвольного $(f(X)) \in R_n$

$$(f(X))(v(X)) \in I. \tag{3.19}$$

Таким образом, из соотношений (3.16) и (3.19) видно, что I является идеалом R_n (см. разд. 2.2).

Пусть $g(X)$ — нормированный многочлен (т. е. многочлен, коэффициент при наивысшей степени X которого равен 1) минимальной степени, не являющийся нулевым и такой, что $(g(X)) \in I$. Пусть $f(X)$ — произвольный многочлен, принадлежащий I. Тогда, если $r(X)$ — остаток от деления $f(X)$ на $g(X)$, то

$$f(X) = a(X)g(X) + r(X)$$

и

$$(r(X)) = (f(X)) - (a(X))(g(X)).$$

Из формул (3.16) и (3.17) следует, что $(r(X)) \in I$. Но тогда, согласно определению $g(X)$, многочлен $r(X)$ должен быть нулевым, так как степень $r(X)$ меньше, чем степень $g(X)$. Это означает, что $f(X)$ делится на $g(X)$. В частности, на $g(X)$ должен делиться двучлен $X^n - 1$, поскольку $(X^n - 1) = (0) \in I$. Обратно, пусть $f(X)$ — произвольный многочлен, который делится на $g(X)$. Тогда, согласно формуле (3.19), $(f(X)) \in I$. Пусть степень $g(X)$ равна l. Так как имеется ровно q^{n-l} многочленов степени $n-1$ и меньше, которые делятся на $g(X)$, то

число классов вычетов, порождаемых идеалом I, равно q^{n-l}. Поскольку соответствие между C и I является взаимно однозначным и C содержит q^k векторов, то $l = n - k$. Далее, если предположить, что существуют два многочлена $g(X)$, удовлетворяющих описанным выше свойствам, то из того, что эти многочлены должны делить друг друга и их старшие коэффициенты равны 1, будет следовать, что эти многочлены совпадают. Это означает, что $g(X)$ определяется однозначно. Таким образом, мы доказали следующую теорему [8, 29—31].

Теорема 3.4. *Пусть C — циклический (n, k)-код и $I = \{(v(X)) \in R_n \mid v \in C\}$. Тогда существует такой нормированный многочлен $g(X)$ степени $n - k$, делящий $X^n - 1$, что выполнение сравнения*

$$f(X) \equiv 0 \qquad (\mathrm{mod}\, g(X))$$

является необходимым и достаточным условием того, что $(f(X)) \in I$.

Многочлен $g(X)$ определяется однозначно и называется *порождающим* многочленом циклического кода C.

Наоборот, пусть $g(X)$ — произвольный нормированный многочлен степени $n - k$, который делит $X^n - 1$. Рассмотрим множество I смежных классов из R_n вида $(a(X))(g(X))$. В каждом смежном классе из I возьмем многочлен $v(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$ степени $n - 1$ или менее, принадлежащий этому смежному классу (такой многочлен только один), и поставим ему в соответствие вектор $(v_0, v_1, \ldots, v_{n-1})$ из V_n. Пусть C — множество всех таких векторов. Легко проверить, что C является циклическим (n, k)-кодом и $g(X)$ — порождающий многочлен этого кода.

Пусть $g(X) = g_0 + g_1X + \ldots + g_{n-k}X^{n-k}$ — порождающий многочлен циклического кода C, и пусть G — матрица следующего вида:

$$G = \begin{array}{c} \longleftarrow n \longrightarrow \\ \begin{bmatrix} g_0 & g_1 & \ldots & g_{n-k} & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \ldots & g_{n-k} & 0 \ldots & 0 \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ \cdot & & \cdot & \cdot & & \cdot & \cdot \\ \cdot & & \cdot & \cdot & & \cdot & 0 \\ 0 & \ldots & g_0 & g_1 & \ldots & & g_{n-k} \end{bmatrix} \end{array} \begin{array}{c} \uparrow \\ k \\ \downarrow \end{array}. \tag{3.20}$$

Так как строки матрицы G — это кодовые слова кода C, а ранг G равен k (поскольку $g_{n-k} = 1$), то матрица G является порождающей матрицей кода C. При этом порождающий многочлен $g(X)$ можно рассматривать как краткое представление

матрицы G. Многочлен $h(X) = (X^n - 1)/g(X)$ называется *проверочным многочленом* кода. Рассмотрим код C_D, двойственный к коду C. В силу того что код C вместе с каждым кодовым вектором v содержит также вектор $P^{n-1}v$, для произвольного вектора $u \in C_D$ $(P^{n-1}v)u^T = 0$, или, что то же самое,

$$v(Pu)^T = 0.$$

Это означает, что $Pu \in C_D$, т. е. код C_D также является циклическим. Порождающий многочлен кода C_D обозначим через $g_D(X)$. Пусть $h(X) = h_0 + h_1X + \ldots + h_kX^k$ и

$$h^* = (h_k, h_{k-1}, \ldots, h_0, \underbrace{0, \ldots, 0}_{n-k-1}).$$

Тогда

$$h^*G^T = \left(\sum_{i=0}^{k} h_{k-i}g_i, \sum_{i=0}^{k-1} h_{k-i-1}g_i, \ldots, h_1g_0\right),$$

где $g_i = 0$, $i > n - k$. Так как $g(X)h(X) = X^n - 1$, то

$$\sum_{i=0}^{k} h_{k-i}g_i = \sum_{i=0}^{k-1} h_{k-i-1}g_i = \ldots = h_1g_0 = 0, \; g_0h_0 = -1.$$

Следовательно,

$$h^*G^T = 0,$$

т. е. $h^* \in C_D$. Многочлен $h_0^{-1}h^*(X)$ должен делиться на $g_D(X)$, но он, так же как и $g_D(X)$, имеет степень k, а его старший коэффициент равен 1. Таким образом,

$$g_D(X) = h_0^{-1}h^*(X) = h_0^{-1}X^kh(X^{-1}).$$

Из вышеизложенного следует справедливость следующей теоремы.

Теорема 3.5. *Двойственный код C_D циклического (n, k)-кода также является циклическим. Порождающим многочленом кода C_D является многочлен $h_0^{-1}X^kh(X^{-1})$, где $h(X)$ — проверочный многочлен кода C, а h_0 — свободный член $h(X)$.*

Из этой теоремы непосредственно следует, что матрица

$$H = \overbrace{\begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & h_0 & \ddots & \vdots \\ \vdots & & \ddots & & & \ddots & \vdots \\ \vdots & & & \ddots & & & 0 \\ 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_0 \end{bmatrix}}^{n} \Big\updownarrow n-k \qquad (3.21)$$

является порождающей матрицей кода C_D и проверочной матрицей кода C.

Поскольку вопросы реализации кодов детально будут рассматриваться в гл. 7, то здесь кратко остановимся лишь на методах кодирования и декодирования, приложимых ко всем циклическим кодам. Условимся, что последние k компонент кодового слова являются информационными символами, а первые $n-k$ компонент — проверочными символами (для конкретности рассмотрим последовательные системы, когда обработка символов ведется последовательно по мере их поступления; случай параллельных систем принципиально не отличается от случая последовательных систем). Представим совокупность информационных символов в виде многочлена $X^{n-k}f_0(X)$ [$f_0(X)$ — многочлен степени $k-1$ или меньше, коэффициентами которого являются информационные символы]. Совокупность проверочных символов представим в виде многочлена $r(X)$ [$r(X)$ — многочлен степени $n-k-1$ или меньше, коэффициентами которого являются проверочные символы]. Согласно теореме 3.4, многочлен $f(X)=$ $=X^{n-k}f_0(X)+r(X)$ является кодовым тогда и только тогда, когда он делится на порождающий многочлен $g(X)$, т. е. тогда и только тогда, когда выполняется следующее равенство $X^{n-k}f_0(X)=a(X)g(X)-r(X)$. Из этого равенства видно, что в качестве $r(X)$ можно взять остаток из деления $X^{n-k}f_0(X)$ на $g(X)$, заменив все его коэффициенты на противоположные. Это отображение $f_0(X)$ в $r(X)$ является линейным и фактически реализует метод нахождения проверочных символов с помощью порождающей матрицы и соотношения (3.4).

Остановимся далее на декодировании. Предположим, что передавался многочлен $f(X)$, а был принят многочлен $f(X)+$ $+e(X)$[1]). Так как $f(X)$ делится на $g(X)$, то, вычислив остаток $r(X)$ от деления $f(X)+e(X)$ на $g(X)$, можно однозначно определить класс вычетов по модулю $g(X)$, которому принадлежит $e(X)$. Из взаимнооднозначности отображений V_n в R_n и C в I, а также описанных выше свойств идеала I легко видеть, что классы вычетов по модулю $g(X)$ соответствуют смежным классам V_n по подгруппе C. При этом многочлены $r(X)$ соответствуют синдромам.

Пороговые методы декодирования и алгебраические методы декодирования БЧХ-кодов будут описаны в следующей главе. В оставшейся части этого раздела рассмотрим ряд других методов декодирования, которые могут быть использованы при небольших длинах кодов и малых скоростях передачи [8, 32—36]. При этом понадобится следующая теорема.

Теорема 3.6. *Пусть C — циклический (n, k)-код с порождающим многочленом $g(X)$ и минимальным весом $d \geqslant 2t+1$. Пусть $e(X)$ — многочлен ошибок веса t или менее, а $r(X)$ — остаток от*

[1]) $e(x)$ — соответствующий вектору ошибок многочлен, который называется многочленом ошибок.

деления $e(X)$ *на* $g(X)$. *Пусть* $r_i(X)$ — *остаток от деления* $X^i r(X)$ *на* $g(X)$, $0 \leqslant i \leqslant n$. *Если вес* $w(r_i(X))$ (*число ненулевых коэффициентов*) *i-го остатка* $r_i(X)$ *меньше или равен* t, *то*

$$e(X) \equiv X^{n-i} r_i(X) \pmod{X^n - 1}.$$

Доказательство. Пусть $e'(X)$ — остаток от деления $X^{n-i} r_i(X)$ на $X^n - 1$. Так как

$$X^i [e(X) - e'(X)] \equiv X^i r(X) - r_i(X) \equiv 0 \pmod{g(X)},$$

то $X^i[e(X) - e'(X)]$, а следовательно, $e(X) - e'(X)$ — кодовый многочлен. Поскольку $e(X)$ и $e'(X)$ имеют вес, меньший чем t, то $e(X) = e'(X)$.

Пусть $t < n/k$ и $e(X)$ — произвольный многочлен, соответствующий вектору ошибок веса t или менее. Тогда выбором подходящего i, $0 \leqslant i < n$, можно добиться того, что

$$X^i e(X) \equiv e_i(X) \pmod{X^n - 1},$$
$$\deg e_i < n - k,$$

где $\deg e_i$ — степень e_i. С другой стороны,

$$e_i(X) \equiv X^i e(X) \equiv X^i r(X) \equiv r_i(X) \pmod{g(X)}.$$

Отсюда следует, что $e_i(X) = r_i(X)$. Вычисляя по $r(X)$ последовательно $r_1(X)$, $r_2(X)$, ..., всегда можно найти такое целое число i, что $w(r_i(X)) \leqslant t$, и, пользуясь предыдущей теоремой, определить $e(X)$. Последовательное вычисление многочленов $r_i(X)$ можно реализовать с помощью регистров сдвига (см. разд. 7.1). В случае когда $t \geqslant n/k$, целое число i, удовлетворяющее указанным выше условиям, может и не существовать. Предположим, что выбором подходящих многочленов $s_0(X) = 0$, $s_1(X)$, $s_2(X)$, ..., $s_l(X)$ можно добиться того, что для любого многочлена ошибок $e(X)$ веса t или меньше будет существовать по крайней мере одна пара целых чисел i и j, такая, что, во-первых, слагаемые $s_j(X)$ с ненулевыми коэффициентами будут входить в качестве слагаемых с ненулевыми коэффициентами также и в $e_i(X)$ и, во-вторых, $\deg(e_i - s_j) < n - k$. Тогда $e_i(X) - s_j(X) = r_i(X) - s'_j(X)$, где $s'_j(X)$ — остаток от деления $s_j(X)$ на $g(X)$. Так как $w(e_i(X) - s_j(X)) \leqslant t$, то $w(r_i(X) - s'_j(X)) \leqslant t$. Обратно, если можно найти такие целые числа i и j, что $w(r_i(X) - s'_j(X)) \leqslant t$, то точно так же, как и при доказательстве теоремы 3.6, можно показать, что

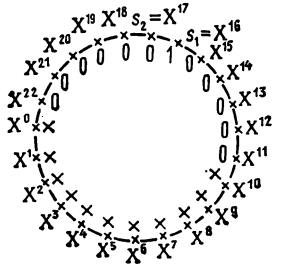
$$X^i e(X) - s_j(X) \equiv r_i(X) - s'_j(X) \pmod{X^n - 1}.$$

В этом случае из сравнения

$$e(X) \equiv X^{n-i} [r_i(X) - s'_j(X) + s_j(X)] \pmod{X^n - 1}$$

находится $e(X)$. Поясним на примере способ выбора многочленов $s_1(X)$, $s_2(X)$, ...

Пример 3.3 [34]. Рассмотрим двоичный циклический код с параметрами $n = 23$, $k = 12$, $t = 3$. Как будет показано ниже в примере 4.2, кодом с такими параметрами является только код Голея. Рассмотрим конфигурации ошибок веса 3. Конфигурации ошибок удобно изображать на окружности длины 23, как показано на фиг. 3.2. Предположим, что на окружности имеются три символа 1, показывающие положение ошибок, и между ними находятся σ_1, σ_2 и σ_3 символов 0 ($\sigma_1 \leqslant \sigma_2 \leqslant \sigma_3$). Если $\sigma_3 \geqslant 12$, то, выбрав надлежащим образом i, можно добиться того, что $\deg e_i < 11$ и $\deg(e_i - s_0) < 11$. Предположим, что $\sigma_3 < 12$.



Фиг. 3.2.

В этом случае $\sigma_1 + \sigma_2 > 8$. Тогда, так как $\sigma_2 \geqslant \sigma_1$, то $\sigma_2 \geqslant 5$. С другой стороны, $\sigma_3 \geqslant 7$. Поэтому по крайней мере один из трех символов 1 обладает тем свойством, что с одной стороны от него располагается по меньшей мере пять символов 0, а с другой — более шести. Тогда, положив $s_1(X) = X^{16}$ и $s_2(X) = X^{17}$, получим, что для любого $e(X)$ веса $w(e(X)) = 3$ существуют такие целые числа i и j ($0 \leqslant i < 23$, $0 \leqslant j < 3$), что

$$\deg(e_i - s_j) < 11.$$

Рассмотрим конфигурации ошибок веса 2. Пусть σ_1 и σ_2 — числа символов 0 между символами 1, определяющими положение ошибок ($\sigma_1 \leqslant \sigma_2$). Если $\sigma_2 \geqslant 12$, то при некотором i

$$\deg(e_i - s_0) < 11.$$

Как легко видеть, в случае, когда $\sigma_2 = 11$, $\sigma_1 = 10$, при некотором i

$$\deg(e_i - s_1) < 11.$$

Если $w(e(X)) = 1$, то при некотором i

$$\deg(e_i - s_0) < 11.$$

Итак, как видно из вышеизложенного, $s_1(X)$ и $s_2(X)$ удовлетворяют нужным условиям. Декодер, основанный на использовании s_0, s_1 и s_2, описан в работах [34, 36].

Утверждение 3.14. *Пусть* $q_1(X) \mid X^n - 1$ *и* $g_1(X)g_2(X) \mid X^n - 1$. *Тогда циклический код* C_2 *длины* n *с порождающим многочленом* $g_1(X)g_2(X)$ *является подкодом циклического кода той же длины с порождающим многочленом* $g_1(X)$.

Краткое доказательство. Так как из того, что $g_1(X)g_2(X) \mid v(X)$, следует, что $g_1(X) \mid v(X)$, то кодовые векторы кода C_2 принадлежат также коду C_1.

Утверждение 3.15. 1) *Если* $g(X)$ *не содержит в качестве сомножителя* $X - 1$, *то циклический код* C *длины* n *с порождающим многочленом* $g(X)$ *содержит кодовое слово* $(1,1, \ldots, 1)$. 2) *Подкод* C_e *кода* C, *задаваемый порождающим многочленом* $(X-1)g(X)$, *является четным подкодом кода* C.

Краткое доказательство. 1) Так как $g(X) \mid X^n - 1$, $(X-1, g(X)) = 1$ и $X^n - 1 = (X-1)(X^{n-1} + X^{n-2} + \ldots + 1)$, то $g(X) \mid X^{n-1} + X^{n-2} + \ldots + 1$. 2) Если $v(X) \in C_e$, то $(X-1) \mid v(X)$. Но тогда $v(1) = 0$. Так как справедливо также и обратное утверждение, то C_e является четным подкодом кода C.

3.6. Циклические коды (II)

Рассмотрим q-ичный циклический (n, k)-код. Порождающий многочлен $g(X)$ этого кода можно разложить единственным образом в произведение нормированных неприводимых многочленов над $GF(q)$. Пусть это разложение имеет вид

$$g_1(X)^{r_1} \ldots g_l(X)^{r_l}.$$

Так как для важнейших кодов $r_1 = r_2 = \ldots = r_l = 1$, то ниже для простоты рассмотрим случай, когда кратные сомножители отсутствуют. Пусть m_i — степень $g_i(X)$ и

$$m = \text{НОК}\ (m_1, m_2, \ldots, m_l),$$

где НОК — обозначает наименьшее общее кратное. Согласно замечанию, сделанному после теоремы 2.13, $g_i(X)$ имеет m_i различных корней в поле $GF(q^{m_i})$, а следовательно, и в $GF(q^m)$. Если β_i — один из этих корней, то $\beta_i^q, \ldots, \beta_i^{q^{m_i-1}}$ — остальные корни $g_i(X)$. Пусть n_i — порядок элемента β_i. Согласно теореме 2.13, показатель $g_i(X)$ равен n_i, так что

$$g_i(X) \mid (X^{n_i} - 1). \tag{3.22}$$

Поскольку $g_i(X) \mid (X^n - 1)$, то из части 2 утверждения 2.54 следует, что $n_i \mid n$. Следовательно,

$$\text{НОК}(n_1, n_2, \ldots, n_l) \mid n.$$

Обозначим НОК $(n_1, n_2, \ldots, n_l)$ через n_0. Из формулы (3.22) и части 2 утверждения 2.54 следует, что

$$g(X) \mid (X^{n_0} - 1).$$

Если $n > n_0$, то вектор $X^{n_0} - 1$ [ниже вектор, соответствующий $(f(X))$, будем называть просто вектором $f(X)$] является кодовым вектором веса 2, так что минимальное расстояние этого кода не больше чем 2, и последний не может исправлять даже одиночные ошибки. Для обнаружения одиночных ошибок можно обойтись, например, кодами с одной общей проверкой на четность. Учитывая это, в дальнейшем всегда будем предполагать, что $n = n_0$.

Покажем, что можно сделать и обратное, а именно задать циклический код, задав корни $g(X)$. Для этого вначале зададим m. Пусть α — один из примитивных элементов поля $GF(q^m)$. Далее зададим l неотрицательных не превосходящих $q^m - 2$ целых чисел $\mu_1, \ldots, \mu_l$, никакие два из которых не принадлежат одному $(q, q^m - 1)$-циклу. Пусть $g_i(X)$ — минимальный многочлен элемента α^{μ_i} с коэффициентами из поля $GF(q)$. Степень m_i многочлена $g_i(X)$ равна длине $(q, q^m - 1)$-цикла, которому принадлежит μ_i. Согласно утверждению 2.13, порядок n_i элемента α^{μ_i} равен $(q^m - 1)/(\mu_i, q^m - 1)$. Тогда в качестве порождающего многочлена циклического кода можно взять произведение $g_1(X)g_2(X) \ldots g_l(X)$. Длина n полученного таким образом кода равна НОК $(n_1, n_2, \ldots, n_l)$, а число информационных символов составляет $n - \sum_{i=1}^{l} m_i$.

Пример 3.4. Пусть $q = 2$ и $m = 4$. Выберем $\mu_1 = 1$ и $\mu_2 = 3$. Как следует из утверждения 2.60, в этом случае $m_1 = m_2 = 4$, $n_1 = 15$, $n_2 = 5$. Но тогда $n = 15$. Используя конкретное представление поля $GF(2^4)$ и элемент α из задачи 2.59, получим, что $g_1(X) = X^4 + X + 1$, $g_2(X) = X^4 + X^3 + X^2 + X + 1$ и $g(X) = g_1(X)g_2(X) = X^8 + X^7 + X^6 + X^4 + 1$. Построенный здесь код является примером БЧХ-кода, исправляющего двойные ошибки (произвольные БЧХ-коды будут описаны в разд. 4.1).

Предположим, что $g(X)$ является произведением различных неприводимых многочленов $g_1(X)$, $g_2(X)$, $\ldots$, $g_l(X)$ с коэффициентами из поля $GF(q)$. Пусть m_i — степень $g_i(X)$, а α_i — один из корней $g_i(X)$. Степень $g(X)$ равна $\sum_{i=1}^{l} m_i$. Поскольку любой

кодовый вектор $v(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$ q-ичного циклического кода C длины n с порождающим многочленом $g(X)$ делится на $g(X)$, то

$$v(\alpha_i) = 0, \quad 1 \leqslant i \leqslant l, \tag{3.23}$$

или, что то же самое,

$$\begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \ldots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \ldots & \alpha_2^{n-1} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 1 & \alpha_l & \alpha_l^2 & \ldots & \alpha_l^{n-1} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \cdot \\ \cdot \\ \cdot \\ v_{n-1} \end{bmatrix} = 0. \tag{3.24}$$

Наоборот, как следует из свойств минимального многочлена, любой вектор $v(X)$, удовлетворяющий указанным выше условиям, делится на $g(X)$, а следовательно, является кодовым вектором. В этом смысле вышеприведенная матрица похожа на проверочную матрицу кода. Элементами этой матрицы являются элементы поля $GF(q^m)$. Однако, используя любой базис поля $GF(q^{m_i})$ каждый элемент α_i^j можно представить в виде вектор-столбца A_{ij}, компонентами которого являются элементы поля $GF(q)$. Заменяя α_i^j на A_{ij}, получаем матрицу размера $r \times n$ с элементами из поля $GF(q)$:

$$M = \begin{bmatrix} A_{10} & A_{11} & \ldots & A_{1\,(n-1)} \\ A_{20} & A_{22} & \ldots & A_{2\,(n-1)} \\ \cdot & \cdot & \cdot & \cdot \\ A_{l_0} & A_{l_1} & \ldots & A_{l\,(n-1)} \end{bmatrix},$$

которая обладает тем свойством, что вектор $v = (v_0, v_1, \ldots, v_{n-1})$ удовлетворяет условию (3.24), т. е. является кодовым тогда и только тогда, когда

$$Mv^T = 0.$$

Следовательно, M является одной из проверочных матриц.

Пример 3.5. Пусть $q = 2$, α — примитивный элемент поля $GF(2^m)$ и $g(X)$ — минимальный многочлен элемента α. Тогда $g(X) \mid X^{2^m-1} - 1$. Пусть C — циклический $(2^m - 1, 2^m - 1 - m)$-код с порождающим многочленом $g(X)$. Согласно равенству (3.24), каждый кодовый вектор $v = (v_0, v_1, \ldots, v_{m-2})$ должен удовлетворять следующему условию:

$$\sum_{i=0}^{2^m-1} v_i\alpha^i = 0. \tag{3.25}$$

Предположим, что вес v меньше или равен 2 и все компоненты v_i вектора v, за исключением v_{i_1} и v_{i_2} $(i_1 < i_2)$, равны нулю. Тогда, так как

$$v_{i_1}\alpha^{i_1} + v_{i_2}\alpha^{i_2} = 0,$$

то

$$v_{i_1} = v_{i_2}\alpha^{i_2 - i_1}.$$

Поскольку $0 < i_2 - i_1 < 2^m - 1$ и α — примитивный элемент поля, то $\alpha^{i_2 - i_1} \neq GF(2)$, так что $v_{i_1} = v_{i_2} = 0$. Таким образом, минимальный вес d кода C не меньше трех. С другой стороны, если i_1 и i_2 — произвольные целые числа, такие, что $0 \leqslant i_1 < i_2 < 2^m - 1$, то $\alpha^{i_1} + \alpha^{i_2} \in GF(2^m)$. Следовательно, существует такое целое число j, $0 \leqslant j < 2^m - 1$, что

$$\alpha^{i_1} + \alpha^{i_2} = \alpha^j.$$

Очевидно, что $j \neq i_1$, i_2. Если v — вектор, компоненты которого v_{i1}, v_{i2} и v_j равны 1, а остальные равны 0, то v удовлетворяет соотношению (3.24) и, следовательно, является кодовым. Таким образом, минимальное расстояние кода C в точности равно 3. Этот код лежит на границе Хэмминга и является совершенным. Код C называется двоичным *циклическим кодом Хэмминга*; коды, эквивалентные C, называют просто двоичными *кодами Хэмминга*. Если в матрице коэффициентов $(\alpha^0, \alpha^1, \ldots, \alpha^{2m-2})$ из (3.25) каждый элемент α^j заменить на вектор-столбец, получающийся транспонированием вектор-строки, который является представлением α^j в базисе 1, α, α^2, ..., α^{m-1}, то получится матрица H размера $m \times (2^m - 1)$ с элементами из поля $GF(2)$, у которой все столбцы различные и ненулевые. Следовательно, утверждение 3.13 дает нумератор весов кода C. Если в случае $m = 3$ взять в качестве минимального многочлена элемента α многочлен $X^3 + X + 1$, то получим

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Изменяя подходящим образом нумерацию компонент кодовых слов, можно получить код из примера 1.4.

Утверждение 3.16. *В примере 3.5 оценка $\alpha \geqslant 3$ была получена на основании равенства (3.25). Однако эту оценку можно также получить исходя из того, что если $g(X) \mid aX^{i_1} + bX^{i_2}$, где $a, b \in GF(2)$, $0 \leqslant i_1 < i_2 < 2^m - 1$, то $a = b = 0$.*

Краткое доказательство. Сначала заметим, что $g(X) \nmid X^i$. Так как $g(X)$ — минимальный многочлен примитивного элемента α, то $g(X)$ является примитивным, а следовательно, $g(X) \nmid X^i - 1$, $0 < i < 2^m - 1$.

Пример 3.6. Пусть $h(X)$ — примитивный многочлен степени m над полем $GF(2)$. Рассмотрим двоичный циклический код длины $n = 2^m - 1$ с порождающим многочленом $g(X) = (X^n + 1)/h(X)$. Так как порождающий многочлен $X^m h(X^{-1})$ двойственного кода также является примитивным многочленом (см. утверждение 2.58), то двойственный код является кодом Хэмминга из примера 3.5. Рассматриваемый код имеет m информационных символов и состоит из 2^m кодовых векторов. Многочлены $g(X)$, $Xg(X)$, ..., $X^{n-1}g(X)$ являются различными кодовыми векторами. В самом деле, допустим, что $X^i g(X) \equiv X^j g(X) \pmod{X^n + 1}$, $0 \leqslant i \leqslant j < n$. Тогда $X^n + 1 \mid X^i g(X)(X^{i-j} + 1)$ и $h(X) \mid X^{j-i} + 1$. Но это противоречит тому, что $h(X)$ — примитивный многочлен. Следовательно, 2^m кодовыми векторами являются нулевой вектор, вектор $g(X)$ и все циклические сдвиги последнего. Проверочным многочленом этого кода является $h(X)$. Если $v = (v_0, v_1, \ldots, v_{n-1})$ — ненулевой кодовый вектор и $h(X) = 1 + h_1 X + h_2 X^2 + \ldots + X^m$, то из (3.21) имеем

$$\sum_{i=0}^{m-1} h_{m-i} v_{j+i} = v_{j+m}, \quad 0 \leqslant j < n, \tag{3.26}$$

где индексы вычисляются по модулю n, а $h_0 = h_m = 1$. Следовательно, если $v_{l+i} = v_{j+i}$ $(0 \leqslant i < m)$ при некоторых l и j, $l \not\equiv j \pmod{n}$, то $v_{l+i} = v_{j+i}$ при всех i, т. е. последовательность $v_0, v_1, \ldots, v_{n-1}, v_0$ является периодической. Однако это противоречит тому, что n векторов, получаемых с помощью последовательных циклических перестановок компонент кодового слова, являются различными. Следовательно, при всех i и j $(i \not\equiv j \pmod{n})$

$$(v_l, v_{i+1}, \ldots, v_{i+m-1}) \neq (v_j, v_{j+1}, \ldots, v_{j+m-1}).$$

Если $(v_i, v_{i+1}, \ldots, v_{i+m-1}) = (0, 0, \ldots, 0)$ при некотором i, то, согласно (3.26), вектор v должен быть нулевым, но по предположению v — ненулевой вектор, поскольку $(v_i, v_{i+1}, \ldots, v_{i+m-1}) \neq 0$ при всех i. Так как число m-мерных ненулевых двоичных векторов над $GF(2)$ равно $n = 2^m - 1$, то для произвольного вектора $(a_0, a_1, \ldots, a_{m-1}) \neq 0$ над $GF(2)$ найдется ровно одно такое целое число i, $0 \leqslant i < n$, что

$$(v_l, v_{l+1}, \ldots, v_{i+m-1}) = (a_0, a_1, \ldots, a_{m-1}).$$

Последовательности $v_0, v_1, \ldots, v_{n-1}, v_0, \ldots$, обладающие этим свойством, называют *последовательностями максимальной длины*. Как следует из вышеизложенного, общее число символов 1, содержащихся в последовательностях $(v_i, v_{i+1}, \ldots, v_{i+m-1})$, $i = 0, 1, \ldots, n-1$, равно общему числу символов 1, содержащихся в ненулевых векторах размерности m над полем $GF(2)$, которое

в свою очередь равно $m2^{m-1}$. Если $v_j = 1$, то число символов 1 среди $v_0, v_1, \ldots, v_{n-1}$ равно 2^{m-1}, а следовательно, вес любого ненулевого кодового слова равен 2^{m-1}. Таким образом, рассматриваемый код является эквидистантным (это можно было бы также показать, используя результаты примера 3.5 и утверждение 3.11). Этот код называют кодом максимальной длины. В разд. 4.3 будет описан алгоритм порогового декодирования кодов максимальной длины.

Для примера рассмотрим случай, когда $m = 3$ и $h(X) = X^3 + X^2 + 1$. При этом $g(X) = (X+1)(X^3 + X + 1)$. Согласно (3.21), матрица H имеет следующий вид:

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

До сих пор в процессе всех построений предполагалось, что выбор одного из примитивных элементов или одного из элементов порядка n уже сделан. Рассмотрим, какое влияние выбор этих элементов оказывает на получающиеся коды. Пусть β и γ — различные элементы порядка n поля $GF(q^m)$. Тогда $1, \beta, \ldots, \beta^{n-1}, \gamma$ — корни уравнения $X^n - 1 = 0$. Так как это уравнение имеет не более чем n корней, то найдется такое целое положительное число ν, что $\gamma = \beta^\nu$ и

$$v(\gamma^i) = \sum_{j=0}^{n-1} v_j \gamma^{ij} = 0 \Leftrightarrow v(\beta^{\nu i}) = \sum_{j=0}^{n-1} v_j \beta^{\nu ij} = 0.$$

Поскольку порядок γ также равен n, то, согласно утверждению 2.13, $(n, \nu) = 1$, так что найдется такое целое положительное число μ, что $\nu\mu \equiv 1 \pmod{n}$. Пусть T_μ — преобразование, переводящее вектор $(v_0, v_1, \ldots, v_{n-1}) \in V_n$ в вектор $(v_0, v_\mu, v_{2\mu}, \ldots, v_{(n-1)\mu})$ (индексы вычисляются по модулю n). Заметим, что это преобразование приводит лишь к перестановке компонент векторов. Если $T_\mu(v_0, v_1, \ldots, v_{n-1}) = v' = (v'_0, v'_1, \ldots, v'_{n-1})$, то $v'_j = v_{\mu j}$ $(0 \leqslant j \leqslant n)$. Следовательно, $v_j = v'_{\nu j}$ $(0 \leqslant j < n)$. Тогда

$$\sum_{j=0}^{n-1} v_j \beta^{\nu ij} = \sum_{j=0}^{n-1} v'_{\nu j} \beta^{i\nu j} = \sum_{j=0}^{n-1} v'_j \beta^{ij}.$$

Таким образом, использование вместо элемента β элемента γ приводит лишь к некоторой перестановке компонент кодовых векторов. Это означает, что построенные в обоих случаях коды эквивалентны, имеют одинаковую весовую структуру, а следовательно, и одинаковую способность исправлять независимые

ошибки. Однако, вообще говоря, эти коды обладают различными способностями исправлять пачки ошибок. Кроме того, описанные в разд. 7.1 кодирующие и декодирующие устройства имеют различные коэффициенты в цепи обратной связи, а значит, и различное число сумматоров.

Утверждение 3.17. *Пусть* $g(X)$ — *многочлен степени* m, *такой, что* $g(X) = X^m g(X^{-1})$. *Тогда, если* C — *циклический код с порождающим многочленом* $g(X)$ *и* $(v_0, v_1, \ldots, v_{n-1}) \in C$, *то* $(v_{n-1}, v_{n-2}, \ldots, v_0) \in C$.

Утверждение 3.18. *Пусть* $v = (v_0, v_1, \ldots, v_{n-1})$ — *кодовое слово* q-*ичного циклического кода* C. *Тогда вектор* $\Pi_q v$, *получающийся из* v *циклической перестановкой компонент* Π_q: $i \to qi \pmod n$, *также принадлежит коду* C.

Краткое доказательство. Если $v(X)$ — представление v в виде многочлена, то $v^q(X)$ — представление $\Pi_q v$ в виде многочлена. Утверждение следует из того, что если $g(X) \mid v(X)$, то $g(X) \mid v^q(X)$.

Методы декодирования циклических кодов, основанные на утверждении 3.18 и теореме 3.6, описаны в работах [33, 37].

3.7. Укороченные коды

Предположим, что линейный (n, k)-код C имеет порождающую матрицу, первые i столбцов которой являются линейно независимыми $(i < k)$. Множество кодовых векторов, первые i компонент которых равны 0, образует подпространство размерности $(k - i)$ пространства (кода) C. Множество C' векторов длины $n - i$, полученных удалением первых i компонент из таких кодовых векторов, образует линейный $(n - i, k - i)$-код. Эта процедура называется укорачиванием, а код C' — укорочением кода C. Минимальное расстояние C' не меньше, чем минимальное расстояние C. В частности, если C — циклический код, то C' называется *укороченным циклическим кодом*. Порождающая матрица G' кода C' получается из порождающей матрицы G вида (3.20) кода C путем удаления первых i строк и i столбцов. Проверочная матрица кода C' получается удалением первых i столбцов из проверочной матрицы кода C. Если вектору $(v_0, v_1, \ldots, v_{n'-1})$ (здесь $n' = n - i$) поставить в соответствие многочлен $v_0 + v_1 X + \ldots + v_{n'-1} X^{n'-1}$ (а не класс вычетов по модулю $X^{n'} - 1$, как в случае циклических кодов), то, как следует из (3.20) и указанного выше способа построения G' [8, 31], $v(X)$ будет кодовым вектором тогда и только тогда, когда $v(X)$ делится на порождающий многочлен $g(X)$ кода C. Так как

в отличие от случая циклических кодов здесь не требуется, чтобы многочлен $g(X)$ делил $X^{n'}-1$, то класс укороченных циклических кодов значительно шире, чем класс циклических кодов. В то же время методы кодирования и декодирования циклических кодов почти без изменения могут использоваться и для укороченных циклических кодов.

Утверждение 3.19 [38]. *Пусть n, r и d — целые числа, удовлетворяющие следующему неравенству*:

$$[(n-1)/r]\sum_{i=0}^{d-2}\binom{n-1}{i}<\phi(r),$$

где $\phi(r)$ — число неприводимых многочленов степени r над полем $GF(2)$. Тогда существует двоичный укороченный циклический код длины n с минимальным расстоянием d или более, порождаемый неприводимым многочленом степени r.

Краткое доказательство. Число многочленов над $GF(2)$ степени не выше $n-1$ со свободным членом, равным 1, и не более чем с $d-1$ ненулевыми коэффициентами равно $\sum_{i=0}^{d-2}\binom{n-1}{i}$. Каждый такой многочлен делится не больше чем на $[(n-1)/r]$ неприводимых многочленов степени r. Следовательно, если приведенное выше соотношение выполняется, то найдется по крайней мере один неприводимый многочлен $g(X)$ степени r, который не делит ни один многочлен ошибок веса $d-1$ или менее. Минимальный вес укороченного циклического кода, порождающим многочленом которого является $g(X)$, больше или равен d.

Утверждение 3.20 [38]. *Неравенство в формулировке предыдущего утверждения можно заменить на неравенство*

$$H(d/n)\leqslant(r/n)(1-O(n)),$$

где $O(n)\to 0$ при $n\to\infty$ и фиксированных отношениях d/n и r/n. Другими словами, при больших n укороченные циклические коды достигают нижней границы Варшамова — Гилберта.

Идея доказательства. Следует воспользоваться верхней границей Чернова и неравенством (2.13).

4. Важнейшие коды

4.1. Коды Боуза — Чоудхури — Хоквингема

Коды Боуза — Чоудхури — Хоквингема [1—4] (БЧХ-коды) составляют один из больших классов кодов, исправляющих независимые ошибки веса t, метод построения которых может быть задан явно. Свойства и методы декодирования этих кодов достаточно подробно исследованы. Пусть m — произвольное целое положительное число и n — один из делителей $q^m - 1$. Согласно теореме 2.6 и утверждению 2.13, в поле $GF(q^m)$ всегда существует элемент порядка n; пусть β — один из таких элементов. Заметим, что все элементы β^i $(0 \leqslant i < n)$ различны. Справедлива следующая теорема.

Теорема 4.1. *Если порождающий многочлен $g(X)$ циклического кода длины n имеет своими корнями $\beta^{l+1}, \beta^{l+2}, \ldots, \beta^{l+r}$ $(1 \leqslant r < n)$, где l $(0 \leqslant l < n)$ — некоторое целое число, то минимальное расстояние этого кода не меньше чем $r+1$.*

Доказательство. Если в (3.24) положить $\alpha_1 = \beta^{l+1}$, $\alpha_2 = \beta^{l+2}, \ldots, \alpha_r = \beta^{l+r}$, то

$$\begin{bmatrix} 1 & \beta^{l+1} & \beta^{2(l+1)} & \ldots & \beta^{(n-1)(l+1)} \\ 1 & \beta^{l+2} & \beta^{2(l+2)} & \ldots & \beta^{(n-1)(l+2)} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & \beta^{l+r} & \beta^{2(l+r)} & \ldots & \beta^{(n-1)(l+r)} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \cdot \\ \cdot \\ \cdot \\ v_{n-1} \end{bmatrix} = 0. \qquad (4.1)$$

Определитель любой подматрицы из r столбцов матрицы в левой части (4.1) имеет следующий вид:

$$\begin{vmatrix} \beta^{j_1(l+1)} & \beta^{j_2(l+1)} & \ldots & \beta^{j_r(l+1)} \\ \beta^{j_1(l+2)} & \beta^{j_2(l+2)} & \ldots & \beta^{j_r(l+2)} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \beta^{j_1(l+r)} & \beta^{j_2(l+r)} & \ldots & \beta^{j_r(l+r)} \end{vmatrix} =$$

$$= \beta^{(j_1+j_2+\ldots j_r)(l+1)r} \begin{vmatrix} 1 & 1 & \ldots & 1 \\ \beta^{j_1} & \beta^{j_2} & \ldots & \beta^{j_r} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \beta^{j_1(r-1)} & \beta^{j_2(r-1)} & \ldots & \beta^{j_r(r-1)} \end{vmatrix}.$$

Определитель в правой части последнего равенства известен как определитель Вандермонда [4, 5], который, как известно, равен $\prod_{u>v} (\beta^{j_u} - \beta^{j_v})$ и в данном случае отличен от нуля, поскольку $\beta^{j_u} \neq \beta^{j_v}$ ($u \neq v$). Отсюда и из первой части теоремы 3.1 следует справедливость теоремы 4.1.

Расстояние $r+1$, гарантируемое этой теоремой, называется *нижней БЧХ-границей* минимального расстояния. Циклические коды с минимальной степенью $g(X)$, или, что то же самое, с минимальным числом проверочных символов, удовлетворяющие указанным в этой теореме условиям при заданных n, l и r, называются *БЧХ-кодами*. Другими словами, БЧХ-код — это циклический код, корнями порождающего многочлена которого являются элементы β^i, где i пробегает все числа из (q, n)-циклов, содержащих $l+1, \ldots, l+r$, и только они. Чтобы теорема 4.1 гарантировала исправление кодом ошибок веса t, следует положить $r = 2t$. Так как, согласно теореме 2.13, длина каждого цикла, содержащего число $l+j$ $(1 \leqslant j \leqslant r)$, равна m или некоторому делителю m, то степень $g(X)$, т. е. число проверочных символов кода, не превосходит $2mt$. Если оба элемента β^l и β^{r+l+1} не являются корнями $g(X)$, то $r+1$ называют *конструктивным расстоянием* данного БЧХ-кода. В частном случае $q = 2$, $n = 2^m - 1$, $l = 0$, $r = 2t$ БЧХ-коды называют также примитивными БЧХ-кодами (или БЧХ-кодами в узком смысле). В случае $t = 1$ эти коды представляют собой не что иное, как *циклические коды Хэмминга*. В табл. А.3, помещенной в приложении, для $m \leqslant 10$ приведены возможные значения конструктивного расстояния БЧХ-кодов, соответствующие им числа информационных символов и, когда это известно, фактическое минимальное расстояние и разность между фактическим и конструктивным минимальными расстояниями. Один из алгебраических методов нахождения числа информационных символов БЧХ-кодов описан в работе [3]. Как показано в примере 4.6, нижняя БЧХ-граница может быть несколько обобщена.

Пример 4.1. Рассмотрим примитивный БЧХ-код с параметрами $m = 4$, $n = 15$, $t = 3$. В этом случае $(2, 15)$-циклами, содержащими числа 1, 2, 3, 4, 5, 6, являются следующие множества: $\{1, 2, 4, 8\}$, $\{3, 6, 12, 9\}$, $\{5, 10\}$. Следовательно, рассматриваемый код имеет 10 проверочных символов. Конструктивное расстояние этого кода равно 7. Выбирая в качестве β корень многочлена $X^4 + X + 1$ и основываясь на задаче 2.60, находим $g(X)$:

$$g(X) = (X^4 + X + 1)(X^4 + X^3 + X^2 + X + 1)(X^2 + X + 1).$$

Пример 4.2. Пусть $n = 23$, $t = 2$, $l = 0$. Минимальное целое число m, такое, что $23 \mid (2^m - 1)$, согласно табл. А.1, равно 11.

Совокупность чисел $\{1, 2, 3, 4\}$ в данном случае содержится в $(2, 23)$-цикле: $\{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$. Следовательно, рассматриваемый код имеет 11 проверочных символов и конструктивное расстояние, равное 5. Если в качестве β взять корень неприводимого многочлена $g(X) = X^{11} + X^9 + X^7 + X^6 + X^5 + X + 1$ с показателем 23, то, как следует из задачи 2.64, $g(X)$ будет порождающим многочленом кода. Вместо $g(X)$ в качестве порождающего многочлена можно также взять $X^{11}g(X^{-1})$. Этот код имеет минимальное расстояние 7 (см. пример 4.11), которое не совпадает с конструктивным расстоянием, равным 5. Прейндж показал, что этот код эквивалентен двоичному совершенному $(23, 11)$-коду Голея [7]. Поскольку $X^{23} - 1 = (X - 1)g(X)X^{11}g(X^{-1})$, то порождающим многочленом кода, двойственного к этому циклическому коду Голея, является $(X - 1)g(X)$. Следовательно, минимальный вес двойственного кода равен 8.

Пример 4.3. Пусть $q = 3$ и β — элемент порядка 11 поля $GF(3^5)$ (так как $3^5 - 1$ делится на 11, то такой элемент существует). В данном случае $(3, 11)$-циклом, содержащим 1, является множество $\{1, 3, 9, 5, 4\}$, нижняя БЧХ-граница равна 4. Однако, как известно [3, 4], минимальное расстояние получающегося кода в действительности равно 5, и этот код эквивалентен троичному $(11, 6)$-коду Голея. Поскольку многочлены $g(X) = X^5 + X^4 + 2X^3 + X^2 + 2$ и $X^5g(X^{-1})$ являются неприводимыми многочленами над $GF(3)$ степени 5 с показателем 11 (см. табл. А.2), то любой из них может быть взят в качестве порождающего многочлена.

Утверждение 4.1. *Пусть m — минимальное целое число, такое, что $n \mid 2^m - 1$. Тогда число проверочных символов двоичного БЧХ-кода длины n с конструктивным расстоянием $2t + 1$ и с $l = 0$ не превосходит mt.*

Краткое доказательство. Числа $2i$ $(0 \leqslant i \leqslant t)$ и i всегда принадлежат одному циклу. Так как длина цикла, содержащего i, равна m или некоторому делителю m, то количество целых положительных чисел, не превосходящих $n - 1$ и входящих в циклы, содержащие числа $1, 2, \ldots, 2t$, не превосходит mt.

Утверждение 4.2. *Пусть $n = (q^m - 1)/(q - 1)$ и $(n, q - 1) = 1$. Пусть β — один из элементов порядка n поля $GF(q^m)$. Если j — такое целое число, что $(q - 1)j \equiv 1 \pmod n$, $0 < j < n$, то порядок β^j равен n, так как $(j, n) = 1$. Циклический код C длины n, порождающим многочленом которого является минимальный многочлен $g(X)$ элемента β^j, является совершенным кодом, исправляющим одиночные ошибки.*

Краткое доказательство. Так как элемент $\beta^{qj} = \beta^{j+1}$ также является корнем $g(X)$, то, согласно теореме 4.1, минимальное

расстояние кода C не меньше 3. Так как этот код лежит на границе Хэмминга, то он является совершенным.

Пример 4.4. БЧХ-коды, для которых $n = q - 1$ и $l = 0$, называются *кодами Рида — Соломона* [8]. Точнее, коды Рида — Соломона — это q-ичные циклические коды длины $q - 1$ с порождающим многочленом $g(X) = \prod_{i=1}^{r}(X - \beta^i)$, где β — один из примитивных элементов поля $GF(q)$. В этом случае минимальное расстояние $d \geqslant r + 1$, а число информационных символов $k = q - 1 - r$. В то же время, согласно следствию 3.1, $d \leqslant r + 1$. Следовательно, для кодов Рида — Соломона

$$d = r + 1.$$

Поскольку соотношение в следствии 3.1 для кодов Рида — Соломона выполняется со знаком равенства, то эти коды являются разделимыми кодами с максимально достижимым расстоянием. Если $g_D(X)$ — порождающий многочлен кода, двойственного к коду Рида — Соломона, то

$$X^{q-1-r} g_D(X^{-1}) = (X^{q-1} - 1)/g(X) = \prod_{i=1}^{q-1}(X - \beta^i) \Big/ \prod_{i=1}^{r}(X - \beta^i) = \prod_{i=r+1}^{q-1}(X - \beta^i),$$

так что $g_D(X) = \prod_{i=0}^{q-2-r}(X - \beta^i)$. Следовательно, минимальное расстояние двойственного кода равно $q - 1 - r$. Коды Рида — Соломона используются для построения кодов Юстесена (разд. 4.6) и кодов, исправляющих пачки ошибок (разд. 4.8).

Пример 4.5. Рассмотрим БЧХ-коды с параметрами $n = q + 1$, $l = -t - 1$, $r = 2t + 1$. Так как $n \mid q^2 - 1$, то $m = 2$. Кроме того, заметим, что $q \equiv -1 \pmod{n}$. Отсюда следует, что $qi \equiv -i \pmod{n}$ для всех i, $0 \leqslant i \leqslant t$. А именно если $i \neq 0$, то $(q, q+1)$-циклом, содержащим i, является множество $\{i, q + 1 - i\}$. Тогда, если β — один из элементов порядка $q + 1$ в поле $GF(q^2)$, то

$$g(X) = \prod_{i=-t}^{i=t}(X - \beta^i)$$

— многочлен над $GF(q)$, являющийся порождающим многочленом рассматриваемого БЧХ-кода. Как и в примере 4.4, из следствия 3.1 получаем, что этот код является разделимым кодом с максимально достижимым расстоянием $d = 2t + 2$. Этот код имеет на два информационных символа больше, чем код Рида — Соломона с теми же параметрами q и d [51].

Задача 4.3. Проверить, что аналогичный результат можно получить, если в примере 4.5 положить $l = (q - 2t + 1)/2$, $q > 2$, и $r = 2t$.

Утверждение 4.4 [4]. *Пусть $q = 2$ и $g_1(X)$ — неприводимый многочлен степени m, удовлетворяющий условию $X^m g_1(X^{-1}) = g_1(X)$. Тогда минимальное расстояние циклического $(n, n - m - 1)$-кода, где n — показатель $g_1(X)$, с порождающим многочленом $g(X) = (X - 1)g_1(X)$ не меньше* 6.

Краткое доказательство. Пусть β — один из корней $g_1(X)$. Так как $q = 2$, то β^2 — также корень $g_1(X)$. Из того, что $X^m g_1(X^{-1}) = g_1(X)$, следует, что β^{-1}, а следовательно, и β^{-2} также будут корнями $g_1(X)$. Таким образом, $g(X)$ имеет своими корнями β^{-2}, β^{-1}, β^0, β^1, β^2, так что, согласно теореме 4.1, минимальное расстояние рассматриваемого кода не меньше 6. По табл. А.2 можно установить, что многочлен $g_1(X) = X^8 + X^7 + X^6 + X^4 + X^2 + X + 1$ является неприводимым и имеет показатель 17. Если рассмотреть циклический $(17, 8)$-код с порождающим многочленом $(X + 1)g_1(X)$, то из верхней границы Хэмминга будет следовать, что минимальный вес этого кода не может быть больше. Следовательно, $d = 6$.

Утверждение 4.5. 1) *Минимальное целое число l, такое, что $2^m + 1 | 2^l - 1$, равно $2m$.* 2) *Минимальное расстояние циклического $(2^m + 1, 2^m + 1 - 2m)$-кода C, порождающим многочленом $g(X)$ которого является неприводимый многочлен степени $2m$ над $GF(2)$ с показателем $2^m + 1$, не меньше* 5.

Краткое доказательство. $2^m + 1 | 2^{2m} - 1$. Если $l < 2m$, то $2^l + 1$ не делится на $2^m + 1$, поскольку $2^l - 1 = 2^{l-m}(2^m + 1) - 2^{l-m} - 1$. Это означает, что существует по крайней мере один неприводимый многочлен, указанный в утверждении 4.5. Если β — один из его корней, то порядок β должен быть равен $2^m + 1$, так что $\beta^{2^m} = \beta^{-1}$. Таким образом, корни $g(X)$ — это элементы $\beta^{\pm 2^i}$, $0 \leqslant i < m$. Следовательно, $X^{2m} g(X^{-1}) = g(X)$. Согласно утверждению 4.4, минимальный вес четного подкода кода C не меньше 6. Если допустить, что минимальный вес кода не больше 4, то вес четного подкода также будет не больше 4, что противоречит доказанному выше.

Конструктивное расстояние является границей снизу для минимального расстояния и, как показывает пример циклического $(23, 11)$-кода Голея, вообще говоря, может не совпадать с минимальным расстоянием. Для примитивных БЧХ-кодов при $m \leqslant 6 (n \leqslant 63)$ эти два параметра совпадают [4], однако минимальное расстояние, равное 31, примитивного $(127, 43)$-БЧХ-кода превышает конструктивное расстояние этого кода, равное 29. Вообще говоря, конструктивное и фактическое мини-

мальное расстояния БЧХ-кодов различаются редко, но тем не менее существует бесконечно много примитивных БЧХ-кодов, минимальное расстояние которых больше конструктивного расстояния (см. табл. А.3) [9]. Известно несколько достаточных условий того, что минимальное и конструктивное расстояния совпадают [3, 4, 10—13]. Одно из таких условий дает приведенное ниже утверждение 4.14. А именно из утверждения 4.14 следует, что минимальное расстояние q-ичного БЧХ-кода длины $n = q^m - 1$ с $l = 0$ и конструктивным расстоянием $(R+1)q^Q - 1$ (где $0 \leqslant Q < m$, $0 \leqslant R < q - 1$) равно конструктивному расстоянию. Рассмотрим произвольный q-ичный БЧХ-код с параметрами $n = q^m - 1$ и $l = 0$. Пусть $d_{\text{БЧХ}}$ — конструктивное расстояние этого кода. Если Δ — минимальное целое число вида $(R' + 1)q^{Q'}$, такое, что $d_{\text{БЧХ}} + 1 \leqslant (R' + 1)q^{Q'}$, то $\Delta/(d_{\text{БЧХ}} + 1) < 2$. Поскольку минимальное расстояние d этого БЧХ-кода не может быть больше минимального веса $\Delta - 1$ БЧХ-кода той же длины с конструктивным расстоянием $(R' + 1)q^{Q'} - 1$, то

$$d \leqslant \Delta - 1 < 2d_{\text{БЧХ}} + 1.$$

Для примитивных кодов последняя граница может быть несколько улучшена [13]:

$$d < 1{,}5d_{\text{БЧХ}} + 0{,}5.$$

Существует гипотеза, что эта оценка является не очень хорошей. Для примитивных БЧХ-кодов при фиксированной скорости $R = k/n$ и $n \to \infty$ имеют место следующие асимптотические равенства: $d/n \approx d_{\text{БЧХ}}/n \approx 2\ln R^{-1}/\log_2 n$ ($\ln$ — обозначение натурального логарифма) [52]. Следовательно, при фиксированной скорости k/n и $n \to \infty$ отношение d/n стремится к нулю.

С помощью описанного в следующем разделе алгебраического алгоритма декодирования можно исправить все конфигурации ошибок, исправление которых гарантируется конструктивным расстоянием (но не минимальным расстоянием). Для бесконечного числа наборов значений параметров в работе [9] показано существование циклических кодов, минимальное расстояние которых несколько больше минимального расстояния БЧХ-кодов с теми же параметрами n и k. Однако, поскольку при длинах n до нескольких тысяч параметры БЧХ-кодов не хуже параметров других кодов, для которых известны конструктивные методы построения, а кроме того, благодаря существованию эффективного алгоритма декодирования БЧХ-коды являются одим из важнейших классов блоковых кодов.

Пример 4.6 [14]. Пусть β — элемент порядка n поля $GF(q^m)$ и C представляет собой q-ичный циклический код длины n с минимальным расстоянием d и порождающим многочленом $g(X)$.

Пусть $g(\beta^{l+i_1+ci_2})=0$, $0\leqslant i_1<d_0-1$, $0\leqslant i_2\leqslant s$, $(n, c)=1$. Тогда

$$d\geqslant d_0+s.$$

Доказательство. Из теоремы 4.1 следует, что $d\geqslant d_0$. Пусть $v(X)$ — кодовый вектор, вес w которого удовлетворяет неравенствам $d_0\leqslant w<d_0+s$. Осуществляя в случае необходимости циклическую перестановку, получаем

$$v(X)=1+\sum_{j=1}^{w-1}Y_jX^{a_j},$$

где $0<a_1<a_2<\ldots<a_{w-1}<n$, $Y_j\in GF(q)-\{0\}$. Положим $X_j=\beta^{a_j}$ и введем величины

$$S_i=\sum_{j=1}^{w-1}Y_jX_j^i.$$

Если $g(\beta^i)=0$, то

$$S_i=-1,$$

поскольку $v(\beta^i)=0$. Кроме того, положим

$$\sigma_1(X)=\prod_{j=1}^{d_0-2}(X-X_j)=\sum_{i_1=0}^{d_0-2}\sigma_{i_1}^{(1)}X^i,$$

$$\sigma_2(X)=\prod_{j=d_0-1}^{w-1}(X-X_j^c)=\sum_{i_2=0}^{w-d_0+1}\sigma_{i_2}^{(2)}X^{i_2},$$

$$\sigma(X)=\sigma_1(X)\,\sigma_2(X).$$

Так как $a_j\neq 0$, $(n, c)=1$, то $X_j\neq 1$ и $X_j^c\neq 1$. Следовательно, $\sigma(1)\neq 0$. С другой стороны,

$$\sum_{j=1}^{w-1}Y_jX_j^l\sigma_1(X_j)\,\sigma_2(X_j^c)=\sum_{j=1}^{m-1}Y_jX_j^l\left(\sum_{i_1=0}^{d_0-2}\sigma_{i_1}^{(1)}X_j^{i_1}\right)\left(\sum_{i_2=0}^{w-d_0+1}\sigma_{i_2}^{(2)}X_j^{ci_2}\right)=$$

$$=\sum_{i_1=0}^{d_0-2}\sum_{i_2=0}^{w-d_0+1}\sigma_{i_1}^{(1)}\sigma_{i_2}^{(2)}S_{l+i_1+ci_2}=0.$$

Так как $S_{l+i_1+ci_2}=-1$, то приведенное выше выражение равно $\sigma(1)$, но это противоречит тому, что $\sigma(1)\neq 0$. Поэтому $d\geqslant d_0+s$. Положив $s=0$, найдем нижнюю БЧХ-границу. Это показывает, что полученная выше граница является обобщением нижней БЧХ-границы. Известен также ряд других обобщений БЧХ-границы [14].

4.2. Декодирование БЧХ-кодов

4.2.1. Основные этапы

Процедура исправления t и менее ошибок для q-ичного циклического кода, корнями порождающего многочлена которого

являются элементы $\beta^{l+1}, \beta^{l+2}, \ldots, \beta^{l+2t}$, где β — элемент порядка n поля $GF(q)$, впервые была предложена Питерсоном для $q = 2$ [15]. Цирлер и Горенстейн [16] обобщили эту процедуру на случай произвольного q.

Пусть $v(X)$ — переданный кодовый вектор, $e(X)$ — вектор ошибок и $r(X) = v(X) + e(X)$ — принятый вектор. Так как $v(\beta^{l+i}) = 0$ $(0 < i \leqslant 2t)$, то

$$r(\beta^{l+i}) = e(\beta^{l+i}), \quad 0 < i \leqslant 2t. \tag{4.2}$$

Вместо того чтобы вычислять эти величины непосредственно по $r(X)$, можно с помощью схемы деления, описанной в разд. 7.1.3, вначале найти остаток от деления $r(X)$ на $g(X)$ и подставить в него элементы β^{l+i}. Вектор ошибок $e(X)$ можно задавать, указывая положение его ненулевых компонент и значения последних. Для обозначения положения ненулевых компонент введем так называемые *локаторы ошибок*, а именно сопоставим j-й компоненте $e(X)$ элемент β^{j-1} поля $GF(q^m)$, который назовем ее локатором. Каждую ненулевую компоненту вектора ошибок будем задавать с помощью пары (X, Y), где $X \in GF(q^m)$ — локатор этой компоненты, а Y — ее значение. Таким образом, если вектор ошибок имеет ν ненулевых компонент, то он может быть задан совокупностью ν пар (X_1, Y_1), (X_2, Y_2), ..., (X_ν, Y_ν), все элементы $X_1, X_2, \ldots, X_\nu$ которых различны, а элементы $Y_1, Y_2, \ldots, Y_\nu$ не равны нулю. В частности, если $q = 2$, то $Y_i = 1$ и для задания вектора ошибок достаточно задать лишь локаторы. Если $X_i = \beta^{\mu_i}$, то в соответствии с определением

$$e(\beta^j) = \sum_{i=0}^{\nu} Y_i \beta^{\mu_i j} = \sum_{i=1}^{\nu} Y_i X_i^j.$$

Положим

$$S_j = \sum_{i=1}^{\nu} Y_i X_i^j, \qquad 0 \leqslant j < n. \tag{4.3}$$

Величины S_j $(l < j \leqslant 2t + l)$ могут быть найдены по принятому вектору с помощью (4.2). Если $S_j = 0$ при всех $l < j \leqslant l + 2t$, то синдром оказывается нулевым и принимается решение об отсутствии ошибок. Если хотя бы одна из указанных выше величин S_j не равна нулю, то принимается решение о том, что

$$1 \leqslant \nu \leqslant t.$$

Дальнейшая задача заключается в том, чтобы каким-либо способом найти пары (X_1, Y_1), (X_2, Y_2), ..., (X_ν, Y_ν) по известным величинам S_j, $l < j \leqslant l + 2t$. Основная идея решения этой задачи состоит в том, что вначале нужно попытаться найти ν, затем локаторы $X_1, \ldots, X_\nu$ и далее по локаторам с помощью фор-

мулы (4.3) найти $Y_1, \ldots, Y_\nu$. Для нахождения локаторов $X_1, \ldots, X_\nu$ введем многочлен

$$\sigma(z) = \prod_{i=1}^{\nu}(1 - X_i z) = \sigma_0 + \sigma_1 z + \sigma_2 z^2 + \ldots + \sigma_\nu z^\nu, \quad (4.4)$$

где $\sigma_0 = 1$ и $\sigma_\nu = \prod_{i=1}^{\nu} X_i \neq 0$. Из формулы (4.4) следует, что

$$Y_i X_i^j \sigma(X_i^{-1}) = \sum_{s=0}^{\nu} \sigma_s Y_i X_i^{j-s} = 0, \qquad 0 \leqslant j < n.$$

Если левые и правые части последнего равенства просуммировать по i от 1 до ν, то, в частности, получим

$$\sigma_\nu S_j + \sigma_{\nu-1} S_{j+1} + \ldots + \sigma_1 S_{j+\nu-1} + S_{j+\nu} = 0, \quad (4.5)$$
$$l < j \leqslant l + 2t - \nu.$$

Матрица коэффициентов

$$M = \begin{bmatrix} S_{l+1} & S_{l+2} & \ldots & S_{l+\nu} \\ S_{l+2} & S_{l+3} & \ldots & S_{l+\nu+1} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ S_{l+\nu} & S_{l+\nu+1} & \ldots & S_{l+2\nu-1} \end{bmatrix},$$

как легко проверить, может быть разложена в произведение трех матриц:

$$M = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ X_1 & X_2 & \ldots & X_\nu \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ X_1^{\nu-1} & X_2^{\nu-1} & \ldots & X_\nu^{\nu-1} \end{bmatrix} \times \begin{bmatrix} Y_1 X_1^{l+1} & 0 & \ldots & 0 \\ 0 & Y_2 X_2^{l+1} & \ldots & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \ldots & Y_\nu X_\nu^{l+1} \end{bmatrix} \times$$
$$\times \begin{bmatrix} 1 & X_1 & \ldots & X_1^{\nu-1} \\ 1 & X_2 & \ldots & X_2^{\nu-1} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 1 & X_\nu & \ldots & X_\nu^{\nu-1} \end{bmatrix}. \quad (4.6)$$

Определители первой и третьей матриц в правой части (4.6) являются определителями Вандермонда. Если все элементы $X_1, \ldots, X_\nu$ различны, а кроме того, $Y_i X_i \neq 0$, $1 \leqslant i \leqslant \nu$, то матрица M не вырождена. Таким образом, задача свелась к тому, что необходимо найти ν. Берлекэмп [3, 4, 17] предложил алгоритм одновременного нахождения ν и решения $\sigma_1, \sigma_2, \ldots, \sigma_\nu$

уравнения (4.5). Если многочлен $\sigma(z)$ найден, то, подставляя в него вместо z последовательно элементы β^{-i}, $0 \leqslant i < n$, можно найти корни $X_1, X_2, \ldots, X_\nu$ уравнения $\sigma(z) = 0$ (для реализации этой процедуры можно воспользоваться методом Ченя, который будет описан в разд. 7.1.3). Далее, решая систему уравнений

$$\sum_{i=1}^{\nu} Y_i X_i^{l+j} = S_{l+j}, \qquad 1 \leqslant j \leqslant \nu,$$

можно найти $Y_1, Y_2, \ldots, Y_\nu$ (для решения этой системы можно воспользоваться методом, предложенным Форни [18]). Определитель матрицы коэффициентов последней системы уравнений является определителем Вандермонда, который в данном случае отличен от нуля. При $q = 2$ последняя операция не нужна; следует лишь заменить символы принятого кодового вектора, соответствующие найденным локаторам ошибок $X_1, X_2, \ldots, X_\nu$, на противоположные.

Если $t/n = \text{const}$ и $n \to \infty$, то число операций при декодировании для описанного выше алгоритма растет пропорционально $n^2 \log_q n$.

Утверждение 4.6. *Матрица*

$$M_l = \begin{bmatrix} S_{l+1} & S_{l+2} & \ldots & S_{l+i} \\ S_{l+2} & S_{l+3} & \ldots & S_{l+i+1} \\ \vdots & \vdots & & \vdots \\ S_{l+i} & S_{l+i+1} & \ldots & S_{l+2i-1} \end{bmatrix}, \qquad \nu \leqslant i \leqslant t,$$

является вырожденной при любом $i > \nu$.

Идея доказательства. По аналогии с выводом (4.6) матрицы M_i можно представить в виде

$$M_l = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ X_1 & X_2 & \ldots & X_\nu \\ \vdots & \vdots & & \vdots \\ X_1^{i-1} & X_2^{i-1} & \ldots & X_\nu^{i-1} \end{bmatrix} \begin{bmatrix} Y_1 X_1^{l+1} & 0 & \ldots & 0 \\ 0 & Y_2 X_2^{l+1} & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & & Y_\nu X_\nu^{l+1} \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & X_1 & \ldots & X_1^{i-1} \\ 1 & X_2 & \ldots & X_2^{i-1} \\ \vdots & \vdots & & \vdots \\ 1 & X_\nu & \ldots & X_\nu^{i-1} \end{bmatrix}.$$

Используя это разложение, можно проверить, что все матрицы M_i, $i > \nu$, являются вырожденными.

Для нахождения числа ошибок ν можно последовательно анализировать матрицы M_t, M_{t-1}, ..., M_i. Как следует из утверждения 4.6, искомое ν равно наибольшему индексу i ($i \leqslant t$), такому, что матрица M_i не вырождена. Этот способ нахождения ν использовали Питерсон [15], а также Горенстейн и Цирлер [16]. Определив таким образом ν, они решали систему уравнений (4.5) и находили значения $\sigma_1, \sigma_2, \ldots, \sigma_\nu$.

Пример 4.7. Рассмотрим (15, 5)-БЧХ-код, исправляющий тройные ошибки, из примера 4.1. Для этого кода $l = 0$ и $t = 3$. В качестве β возьмем корень примитивного многочлена $X^4 + X + 1$. Предположим, что ошибки возникли в 1, 3 и 9-й компонентах. Тогда локаторами являются β^0, β^2, β^8. Используя таблицу из задачи 2.59, получаем

$$S_1 = r(\beta) = 1 + \beta^2 + \beta^8 = 0, \quad S_2 = S_1^2 = 0,$$

$$S_3 = r(\beta^3) = 1 + \beta^6 + \beta^9 = \beta^{10}, \quad S_4 = S_2^2 = 0,$$

$$S_5 = r(\beta^5) = 1 + \beta^{10} + \beta^{10} = 1, \quad S_6 = S_3^2 = \beta^5.$$

Находим $M_t = M_3$:

$$M_3 = \begin{bmatrix} 0 & 0 & \beta^{10} \\ 0 & \beta^{10} & 0 \\ \beta^{10} & 0 & 1 \end{bmatrix}.$$

Так как $\det M_3 \neq 0$, то, согласно лемме 4.6, $\nu = 3$. Система уравнений (4.5), в данном случае принимающая вид

$$\beta^{10}\sigma_1 = 0,$$

$$\beta^{10}\sigma_2 = 1,$$

$$\beta^{10}\sigma_3 + \sigma_1 = \beta^5,$$

имеет следующее решение: $\sigma_1 = 0$, $\sigma_2 = \beta^{-10} = \beta^5$, $\sigma_3 = \beta^{-5} = \beta^{10}$. Следовательно,

$$\sigma(z) = 1 + \beta^5 z + \beta^{10} z^2.$$

С помощью таблицы из задачи 2.59 можно проверить, что корнями этого уравнения являются 1, $\beta^{-2} = \beta^{13}$ и $\beta^{-8} = \beta^7$. Рассмотренный код совпадает с кодом Рида — Маллера первого порядка длины 15, для которого известен простой алгоритм порогового декодирования (утверждение 4.23).

Задача 4.7. Рассмотреть декодирование двоичного циклического кода Хэмминга.

Решение. Так как $t = 1$, то $\nu = 1$, если только $e(\beta) \neq 0$. В этом случае $e(\beta)$ сразу определяет локатор. Поскольку код Хэмминга является вырожденным кодом, исправляющим пачки ошибок (длина пачки ошибок равна 1), то, кроме того, он может быть декодирован так, как описано в разд. 4.7.

4.2.2. Итеративный алгоритм Берлекэмпа [1])

Опишем итеративный алгоритм Берлекэмпа в модификации, предложенной Месси [17]. Пусть $s_0, s_1, \ldots, s_{N-1}$ — последовательность элементов поля F. Последовательность $c_1, c_2, \ldots, c_L$ элементов того же поля, удовлетворяющую условию

$$s_j + \sum_{i=1}^{L} c_i s_{j-i} = 0, \qquad L \leqslant j < N, \quad c_i \in F, \quad 1 \leqslant i \leqslant L, \quad (4.7)$$

назовем порождающей последовательностью длины L для последовательности $s_0, s_1, \ldots, s_{N-1}$. Порождающую последовательность минимальной длины будем называть минимальной порождающей последовательностью.

Утверждение 4.8. *Пусть* $s_i = 0$ $(0 \leqslant i < u_0 - 1 < N)$ *и* $s_{u_0-1} \neq 0$. *При* $u_0 = N$ *порождающей последовательности не существует. При* $u_0 = N - 1$ *порождающая последовательность существует и ее минимальная длина равна* $N - 1$.

Идея доказательства. В случае если $u_0 = N$, то при $j = N - 1$ равенство (4.7) не может иметь места. Если же предположить, что $u_0 = N - 1$ и $L < N - 1$, то равенство (4.7) при $j = N - 2$ не справедливо ни при каких $c_1, \ldots, c_{N-2}$. В то же время, полагая $c_1 = -s_{N-1}/s_{N-2}$, $c_i = 0$ $(2 \leqslant i < N)$, получаем порождающую последовательность длины $L = N - 1$.

Следующая лемма устанавливает связь между последовательностями $S_{l+1}, S_{l+2}, \ldots, S_{l+2t}$ и $\sigma_1, \sigma_2, \ldots, \sigma_\nu$, введенными в разд. 4.2.1.

Лемма 4.1. *Последовательность* $\sigma_1, \sigma_2, \ldots, \sigma_\nu$ *является единственной минимальной порождающей последовательностью для последовательности* $S_{l+1}, S_{l+2}, \ldots, S_{l+2t}$.

Доказательство. Положим $s_j = S_{l+1+j}$, $0 \leqslant j < 2t$, $N = 2t$. Из формул (4.5) и (4.7) следует, что $\sigma_1, \sigma_2, \ldots, \sigma_\nu$ — порождающая последовательность для последовательности $S_{l+1}, S_{l+2}, \ldots, S_{l+2t}$. Предположим, что существует другая порождающая последовательность $\sigma'_1, \sigma'_2, \ldots, \sigma'_L$ длины L, меньшей или рав-

[1]) Для поля действительных чисел аналогичный алгоритм был найден ранее Тренчем [19].

ной ν. Тогда

$$S_j + \sum_{i=1}^{L} \sigma_i' S_{j-i} = 0, \qquad l + L < j \leqslant l + L + \nu.$$

Однако это противоречит тому, что матрица M является невырожденной.

Наша задача сейчас состоит в том, чтобы получить эффективный алгоритм нахождения минимальной порождающей последовательности. Если существует минимальная порождающая последовательность для префикса $s_0, s_1, \ldots, s_{u-1}$ $(0 \leqslant u < N)$ заданной последовательности $s_0, s_1, \ldots, s_{N-1}$, то будем обозначать ее через $c_1^{(u)}, c_2^{(u)}, \ldots, c_{L(u)}^{(u)}$. Для удобства представим эту последовательность коэффициентов в виде многочлена

$$c^{(u)}(D) = 1 + c_1^{(u)} D + \ldots + c_{L(u)}^{(u)} D^{L(u)}.$$

Если порождающая последовательность не существует, будем полагать

$$L(u) = u.$$

По определению $L(u+1) \geqslant L(u)$.

Лемма 4.2 [17]. *Если* $s_u + \sum_{i=1}^{L(u)} c_i^{(u)} s_{u-i} \neq 0$, *то*

$$L(u+1) \geqslant u + 1 - L(u).$$

Доказательство. Предположим, что

$$L(u) \leqslant u - L(u). \tag{4.8}$$

По определению

$$\sum_{i=1}^{L(u)} c_i^{(u)} r_{j-i} = -s_j, \qquad L(u) \leqslant j < u, \tag{4.9}$$

$$\sum_{h=1}^{L(u+1)} c_h^{(u+1)} s_{j-h} = -s_j, \qquad L(u+1) \leqslant j < u+1. \tag{4.10}$$

Отсюда и из формул (4.8) — (4.10) получаем

$$\sum_{i=1}^{L(u)} c_i^{(u)} s_{u-i} = -\sum_{i=1}^{L(u)} c_i^{(u)} \sum_{h=1}^{L(u+1)} c_h^{(u+1)} s_{u-i-h} =$$

$$= -\sum_{h=1}^{L(u+1)} c_h^{(u+1)} \sum_{i=1}^{L(u)} c_i^{(u)} s_{u-i-h} = \sum_{h=1}^{L(u+1)} c_h^{(u+1)} s_{u-h} = -s_u,$$

но это противоречит предположению.

Основываясь на вышеизложенном, рассмотрим следующий итеративный алгоритм построения минимальной порождающей последовательности для последовательности $s_0, s_1, \ldots, s_{N-1}$.

Обозначим через α_u подпоследовательность $s_0, s_1, \ldots, s_{u-1}$ последовательности $s_0, s_1, \ldots, s_{N-1}$. Можно, очевидно, ограничиться рассмотрением случая, когда среди символов s_i, $0 \leqslant i < N$, имеется хотя бы один, отличный от нуля. Пусть $u_0 - 1$ — минимальное значение i, такое, что $s_i \neq 0$. Из утверждения 4.8, следует, что

1) если $u_0 = N$, то порождающей последовательности не существует;

1′) если $u_0 < N$, то для α_{u_0+1} существует минимальная порождающая последовательность длины u_0, а именно

$$c^{(u_0+1)}(D) = 1 - s_{u_0} s_{u_0-1}^{-1} D.$$

В этом случае

$$L(u_0 + 1) = u_0.$$

Для удобства положим

$$L(u_0 - 1) = 0, \quad L(u_0) = u_0. \tag{4.11}$$

2) Предположим, что для α_j, $u_0 < j \leqslant u < N$, найдена порождающая последовательность $c^j(D)$. Введем следующие величины:

$$\delta_{u_0-1} = s_{u_0-1},$$

$$\delta_j = s_j + \sum_{i=1}^{L(j)} c_i^{(j)} s_{j-i}, \qquad u_0 < j \leqslant u.$$

Кроме того, предположим, что

$$L(j) = \max [L(j-1),\ j - L(j-1)], \tag{4.12}$$

если $u_0 < j \leqslant u < N$ и $\delta_{j-1} \neq 0$. Справедливость равенства (4.12) при $j = u_0$, $u_0 + 1$ очевидна. Так как $L(u_0 - 1) < L(u_0) = u_0 = L(u_0 + 1) \leqslant L(u)$, то существует целое $\bar{u}$, такое, что

$$L(\bar{u}) < L(\bar{u} + 1) = L(u), \quad u_0 - 1 \leqslant \bar{u} < u. \tag{4.13}$$

При этом либо $\bar{u} = u_0 - 1$, либо $\bar{u} > u_0$. Если $\bar{u} = u_0 - 1$, то $\delta_{\bar{u}} = s_{u_0-1} \neq 0$. Допустим, что $\delta_{\bar{u}} = 0$, $\bar{u} > u_0$. Тогда $c_0^{(\bar{u})}, c_1^{(\bar{u})}, \ldots, c_{L(\bar{u})}^{(\bar{u})}$ — порождающая последовательность для $\alpha_{\bar{u}+1}$, но это противоречит (4.13). Следовательно,

$$\delta_{\bar{u}} \neq 0.$$

Лемма 4.3. *Многочлен*

$$c^{(u+1)}(D) = c^{(u)}(D) + \delta_u \delta_{\bar{u}}^{-1} D^{u-\bar{u}} c^{(\bar{u})}(D), \tag{4.14}$$

где $c^{(u_0-1)}(D) = 1$, *определяет порождающую последовательность для* α_{u+1}. *Если* $c^{(u+1)}(D) \neq c^{(u)}(D)$, *то* $L(u+1) = \max [L(u), u + 1 - L(u)]$.

Доказательство. Согласно соотношениям (4.12) и (4.13),

$$L(u) = L(\bar{u}+1) = \bar{u}+1-L(\bar{u}). \tag{4.15}$$

Если $\delta_u = 0$, то $c^{(u+1)}(D) = c^{(u)}(D)$, $L(u+1) = L(u)$. Если $\delta_u \neq 0$, то, полагая

$$L = \max[L(u),\ u+1-L(u)], \tag{4.16}$$

из формулы (4.15) получаем, что $L = \max[L(u),\ u-\bar{u}+L(\bar{u})]$. Из (4.14) в свою очередь следует, что

$$s_j + \sum_{i=1}^{L} c_i^{(u+1)} s_{j-i} = s_j + \sum_{i=1}^{L(u)} c_i^{(u)} s_{j-i} - \delta_u \delta_{\bar{u}}^{-1} \left[s_{j-u+\bar{u}} + \sum_{j=1}^{L(\bar{u})} c_i^{(\bar{u})} s_{j-u+\bar{u}-i} \right], \tag{4.17}$$

где $L \leqslant j \leqslant u$. Из определения $c^{(u)}(D)$ и того, что $L(u) \leqslant L$, имеем

$$s_j + \sum_{i=1}^{L(u)} c_i^{(u)} s_{j-i} = \begin{cases} 0, & L \leqslant j < u, \\ \delta_u, & j = u. \end{cases} \tag{4.18}$$

Если j меняется между L и u, то $j-u+\bar{u}$ меняется от $L-u+\bar{u}$ до $\bar{u}$. Тогда, так как $L-u+\bar{u} \geqslant \bar{u}+1-L(u) = L(\bar{u})$ (это следует из (4.15) и (4.16)), то $L(\bar{u}) \leqslant j-u+\bar{u} \leqslant \bar{u}$. Если $\bar{u} > u_0$, то, согласно определению $c^{(\bar{u})}(D)$,

$$s_{j-u+\bar{u}} + \sum_{i=1}^{L(\bar{u})} c_i^{(\bar{u})} s_{j-u+\bar{u}-i} = \begin{cases} 0, & L \leqslant j < u, \\ \delta_{\bar{u}}, & j = u. \end{cases} \tag{4.19}$$

Поскольку $s_i = 0$ $(0 \leqslant i < u_0 - 1)$, это равенство справедливо также при $\bar{u} = u_0 - 1$. Из формул (4.17)—(4.19) получаем

$$s_j + \sum_{i=1}^{L} c_i^{(u+1)} s_{j-i} = 0, \qquad L \leqslant j < u+1,$$

т. е. $c^{(u+1)}(D)$ задает порождающую последовательность длины L для α_{u+1}. Согласно лемме 4.2 и равенству (4.16), длина L является минимальной и определяется равенством

$$L(u+1) = \max[L(u),\ u+1-L(u)].$$

Из этой леммы следует, что $c^{(N)}(D)$ определяет минимальную порождающую последовательность для $s_0, s_1, \ldots, s_{N-1}$.

Пример 4.8. Рассмотрим случай исправления трех ошибок кодом из примера 4.7. Здесь

$$(S_1, S_2, \ldots, S_6) = (s_0, s_1, \ldots, s_5) = (0, 0, \beta^{10}, 0, 1, \beta^5).$$

Сначала положим $u_0 = 3$. Из утверждения 1′ имеем

$$c^{(4)}(D) = 1,$$
$$L(4) = 3.$$

При этом, согласно соотношению (4.11),

$$L(2) = 0, \quad L(3) = 3.$$

Поэтому если $u = 4$, то $\bar{u} = 2$. В то же время, так как $\delta_2 = \beta^{10}$ и $\delta_4 = 1$, то из формулы (4.14) и равенства $c^{(2)}(D) = 1$ имеем

$$c^{(5)}(D) = 1 - \beta^{-10}D^2c^{(2)}(D) = 1 - \beta^5 D^2,$$
$$L(5) = \max[L(4),\ 5 - L(4)] = 3.$$

Далее, если $u = 5$, то $\bar{u} = 2$ и

$$\delta_5 = s_5 + \sum_{i=1}^{3} c_i^{(5)} s_{5-i} = \beta^5 - \beta^5 \cdot 0 = \beta^5.$$

Из формулы (4.14) имеем

$$c^{(6)}(D) = c^{(5)}(D) - \beta^{-10}\beta^5 D^3 c^{(2)}(D) = 1 - \beta^5 D^2 - \beta^{10} D^3,$$
$$L(6) = \max[L(5),\ 6 - L(5)] = 3.$$

Таким образом, $\sigma_1 = 0$, $\sigma_2 = \beta^5$, $\sigma_3 = \beta^{10}$.

В случае $q = 2$, используя вместо системы уравнений (4.5) равенства Ньютона [20], связывающие величины $S_l = \sum_{i=1}^{\nu} X_i^l$ ($l < j \leqslant l + 2t$) с элементарными симметрическими функциями $X_1 + X_2 + \ldots + X_\nu = +\sigma_1$, $X_1X_2 + X_2X_3 + \ldots + X_{\nu-1}X_\nu = \sigma_2$, ..., $X_1X_2 \ldots X_\nu = (-1)^\nu \sigma_\nu$ переменных $X_1, X_2, \ldots, X_\nu$, можно определить $\sigma_1, \sigma_2, \ldots, \sigma_\nu$ по S_j ($l < j \leqslant l + 2t$). Питерсон [15] использовал именно этот метод. В этом случае можно применить такой же рекуррентный метод решения, как и ранее, и снизить таким образом число шагов с $2t$ до t [3, 4].

Методы декодирования, описанные в разд. 4.2.1 и 4.2.2, предназначены для исправления ошибок, вес которых не превосходит $(d-1)/2$, где d — конструктивное расстояние даже в том случае, если фактическое минимальное расстояние кода больше конструктивного. Были разработаны алгоритмы декодирования ошибок более высокой кратности, но их эффективность оказалась не очень высокой [21].

4.3. Методы мажоритарного декодирования

В некоторых случаях оказывается более предпочтительным использовать коды, корректирующая способность которых несколько хуже, чем у лучших известных кодов, но которые могут

быть просто реализованы. Типичными представителями таких кодов являются *коды, допускающие мажоритарное декодирование* (см. разд. 4.5.2). В некоторой области значений параметров эти коды имеют корректирующую способность, незначительно уступающую корректирующей способности БЧХ-кодов, но зато их можно проще реализовать [3, 4, 22—24].

Определим *q-ичную мажоритарную функцию* $Maj(x_1, x_2, \ldots, x_m)$ от m переменных $x_1, x_2, \ldots, x_m$, принимающих значения в $GF(q)$, следующим образом:

1) Если некоторые $l > m/2$ переменных $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ принимают одно и то же значение a, то значение мажоритарной функции также равно a.

2) Во всех остальных случаях мажоритарная функция принимает нулевое значение.

При $q = 2$ и нечетных m введенная функция является обычной мажоритарной функцией и может быть реализована с помощью порогового элемента. В случае четных m предпочтение отдается символу 0.

Вначале поясним метод порогового декодирования на простом примере.

Пример 4.9. Рассмотрим (7, 3)-код максимальной длины. Матрица H в данном случае имеет следующий вид:

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Пусть $e = (e_0, e_1, \ldots, e_6)$ — вектор ошибок, а s_0, s_1, s_2 и s_3 — компоненты синдрома. Тогда

$$s_0 = e_0 + e_1 + e_3,$$
$$s_1 = e_1 + e_2 + e_4,$$
$$s_2 = e_2 + e_3 + e_5,$$
$$s_3 = e_3 + e_4 + e_6.$$

Положив $s'_0 = s_0$, $s'_1 = s_0 + s_1 + s_2$ и $s'_2 = s_0 + s_1 + s_3$, получаем

$$s'_0 = e_0 + e_1 + e_3,$$
$$s'_1 = e_0 + e_4 + e_5,$$
$$s'_2 = e_0 + e_2 + e_6.$$

Заметим, что каждая из компонент e_i вектора ошибок, за исключением e_0, входит только в одну из сумм s'_0, s'_1, s'_2. Предположим, что произошла только одна ошибка. Тогда, если $e_0 = 0$, то одна из сумм s'_0, s'_1 и s'_2 будет равна 1, а остальные две будут равны 0. Если $e_0 = 1$, то $s'_0 = s'_1 = s'_2 = 1$. Следовательно, при отсутствии ошибок, а также при одиночных ошибках

$$e_0 = Maj(s'_0, s'_1, s'_2) = Maj(s_0, s_0 + s_1 + s_2, s_0 + s_1 + s_3).$$

Так как рассматриваемый код является циклическим, то аналогичные соотношения имеют место и для остальных i:

$$e_i = Maj(s_i, s_i + s_{i+1} + s_{i+2}, s_i + s_{i+1} + s_{i+3}),$$

где индексы вычисляются по модулю 7. Таким образом, вычисляя s_0, s_1, s_2, s_3 и осуществляя их сдвиги, с помощью вышеприведенной формулы можно провести декодирование. Вопросы технической реализации этой процедуры декодирования будут рассмотрены детально в разд. 7.1.3в.

Пусть C — q-ичный линейный (n, k)-код и пусть C_D — код, двойственный к C. Обозначим через $y = (y_0, y_1, \ldots, y_{n-1})$ и $e = (e_0, e_1, \ldots, e_{n-1})$ соответственно принятый вектор и вектор ошибок. Тогда для любого вектора $a = (a_0, a_1, \ldots, a_{n-1}) \in C_D - \{0\}$ выполняется равенство $\sum_{j=0}^{n-1} a_j y_j = \sum_{j=0}^{n-1} a_j e_j$; последние суммы называются *проверочными суммами*. Пусть линейные комбинации $s_i = \sum_{j=0}^{n-1} a_{ij} e_j$ $(a_{ij} \in GF(q),\ 1 \leqslant i \leqslant l)$ символов e_0, e_1, $\ldots$, e_{n-1} (ниже эти линейные комбинации называются также линейными e-суммами) и непустое подмножество J множества $\{0, 1, \ldots, n-1\}$ удовлетворяют следующим двум условиям:

1) для любого $j \in J$ существует такое $a_j \in GF(q) - \{0\}$, что

$$a_{ij} = a_j, \qquad 0 \leqslant i \leqslant l;$$

2) для любого $j \notin J$ все символы $a_{1j}, a_{2j}, \ldots, a_{lj}$, за исключением, быть может, одного, равны 0. Такие линейные e-суммы $s_1, s_2, \ldots, s_l$ называются *ортогональными* относительно $\sum_{j \in J} a_j e_j$. Линейные e-суммы s'_0, s'_1 и s'_2 из примера 4.9 ортогональны относительно e_0.

Вес вектора e будем, как обычно, обозначать через $w(e)$.

Лемма 4.4. *Если линейные e-суммы $s_1, s_2, \ldots, s_l$ ортогональны относительно линейной e-суммы s и $w(e) \leqslant [l/2]$, то*

$$s = Maj(s_1, s_2, \ldots, s_l). \tag{4.10}$$

Доказательство. Пусть $s = \sum_{j \in J} a_j e_j$, $a_j \neq 0$ $(j \in J)$. Если

$e_j = 0$ $(j \in J)$, то, согласно условию 2 ортогональности, среди $s_1, s_2, \ldots, s_l$ самое большее $[l/2]$ линейных e-сумм отличны от нуля. В этом случае справедливость соотношения (4.20) непосредственно следует из определения Maj. Если $e_j \neq 0$ хотя бы для одного $j \in J$, то количество компонент $e_i \neq 0$, $i \notin J$, не превышает $[l/2] - 1$. Но тогда из определения ортогональности получаем, что не меньше чем $l + 1 - [l/2]$ линейных e-сумм среди $s_1, s_2, \ldots, s_l$ равны s. Таким образом, в этом случае равенство (4.20) также выполняется.

Если в лемме 4.4 $s = e_j$, то формула (4.20) принимает следующий вид:

$$e_j = Maj(s_1, s_2, \ldots, s_l). \tag{4.21}$$

Если линейные e-суммы $s_1, s_2, \ldots, s_l$ являются проверочными, то после того, как по принятому вектору y найдены значения $s_1, s_2, \ldots, s_l$, с помощью вышеприведенной формулы можно определить значение e_j. Если для каждого $0 \leqslant j < n$ существуют проверочные суммы $s_{j1}, s_{j2}, \ldots, s_{j,2t}$, ортогональные относительно e_j, то говорят, что *ошибки веса до t можно исправить с помощью порогового декодирования в один шаг.* Если код является циклическим, то достаточно потребовать существования ортогональных проверочных сумм только для e_0.

Указанную выше идею порогового декодирования в один шаг можно развить дальше следующим образом. Выше предполагалось, что $s_{j1}, s_{j2}, \ldots, s_{j,2t}$ являются проверочными суммами. Здесь же будем считать, что s_{ji} — это просто линейные e-суммы, для каждой из которых, однако, существуют $2t$ проверочных сумм $s_{ji1}, s_{ji2}, \ldots, s_{ji,2t}$, ортогональных относительно s_{ji}. Тогда, если $w(e) \leqslant t$, то, вычисляя по принятому вектору y значения всех проверочных сумм s_{jih}, с помощью леммы 4.4 можно найти значения всех линейных e-сумм s_{ji} и, далее, вновь обращаясь к лемме 4.4, найти e_j. В этом случае говорят, что *ошибки веса до t можно исправить с помощью порогового декодирования в 2 шага.* Совершенно аналогично можно описать декодирование в 3, 4, ..., L шагов. В последнем случае говорят, что *ошибки веса до t включительно можно исправить с помощью порогового декодирования в L шагов.*

Этот метод первоначально был предложен Ридом [25] для декодирования кодов Рида — Маллера (см. разд. 4.5). Имеет место следующая теорема [4, 24].

Теорема 4.2. *Пусть код C допускает исправление ошибок веса до t_1 с помощью порогового декодирования в один шаг. Тогда, если код C_D, двойственный к коду C, имеет минимальное расстояние d_D, то*

$$t_1 \leqslant \frac{n-1}{2(d_D - 1)}.$$

Доказательство. По предположению существуют $2t_1$ проверочных сумм $s_{01}, s_{02}, \ldots, s_{0,\,2t_1}$, ортогональных относительно e_0. Пусть $a^{(i)} = (a_0^{(i)}, a_1^{(i)}, \ldots, a_{n-1}^{(i)})$ — вектор, компонентами которого являются коэффициенты проверочной суммы s_{0i}. По предположению $a_0^{(i)} \neq 0$. Если вес $a^{(i)}$ равен 1, то $d_D = 1$, поскольку $a^{(i)} \in C_D$. Следовательно, в этом случае теорема справедлива. Предположим, что $d_D > 1$. Тогда каждый вектор $a^{(i)}$ содержит по крайней мере $(d_D - 1)$ компонент $a_j^{(i)} \neq 0$, $j \neq 0$. С другой стороны, при каждом $0 < j < n$ из символов $a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{2t_1}$ только один может быть ненулевым. Отсюда следует, что $2(d_D - 1)t_1 \leqslant n - 1$.

Из этой теоремы видно, что возможности порогового декодирования в один шаг очень ограничены. Например, как следует из примера 4.2, в случае двоичного $(23, 12)$-кода Голея $d_D = 8$, так что $t_1 \leqslant 22/14$. Таким образом, $(23, 12)$-код Голея, который, вообще говоря, позволяет исправлять ошибки веса 3 и менее, при пороговом декодировании в один шаг позволяет исправлять только одиночные ошибки. Это же можно сказать и о кодах Рида — Соломона (см. пример 4.4). Если минимальное расстояние кода равно d и он позволяет исправлять ошибки веса до $[(d-1)/2]$ с помощью порогового декодирования в L шагов, то говорят, что код допускает *полную ортогонализацию* в L шагов. Для произвольного L имеет место следующая теорема.

Теорема 4.3. *Пусть код C допускает исправление ошибок веса t_L и менее с помощью порогового декодирования в L шагов. Если d_D — минимальное расстояние кода, двойственного к коду C, то* [1])

$$t_L \leqslant \frac{n - (d_D/2)}{2[d_D/2]}.$$

Доказательство. По предположению существуют $2t_L$ проверочных сумм $s_1, s_2, \ldots, s_{2t_L}$, ортогональных относительно некоторой линейной e-суммы s. Пусть $a, a^{(1)}, \ldots, a^{(2t_L)}$ — векторы, компонентами которых являются коэффициенты сумм $s, s_1, \ldots, s_{2t_L}$. Без ограничения общности можно предположить, что $w(a^{(1)} - a) \leqslant w(a^{(i)} - a)$, $0 < i \leqslant 2t_L$. Из определения ортогональности следует, что при любом $0 \leqslant j < n$ только одна из компонент с номером j векторов $a, a^{(1)} - a, a^{(2)} - a, \ldots, a^{2t_L} - a$

[1]) Через $[A]$ здесь обозначено наибольшее целое число, меньшее или равное A (целая часть A), а через $]A[$ — наименьшее целое число, большее или равное A.

может быть отличной от нуля. Следовательно,

$$w(a) + \sum_{i=1}^{2t_L} w(a^{(i)} - a) \leqslant n, \tag{4.22}$$

$$w(a^{(1)}) = w(a) + w(a^{(1)} - a), \tag{4.23}$$

$$w(a^{(i)} - a^{(1)}) = w(a^{(i)} - a) + w(a^{(1)} - a), \quad 1 < i \leqslant 2t_L. \tag{4.24}$$

Так как $a^{(1)}$, $a^{(i)} - a^{(1)} \in C_D$, то

$$w(a^{(1)}) \geqslant d_D, \tag{4.25}$$

$$w(a^{(i)} - a^{(1)}) \geqslant d_D, \quad 1 < i \leqslant 2t_L. \tag{4.26}$$

С другой стороны, по предположению $w(a^{(1)} - a) \leqslant w(a^{(i)} - a)$. Из формул (4.24) и (4.26) имеем

$$w(a^{(i)} - a) \geqslant [d_D/2], \quad 1 < i \leqslant 2t_L.$$

Отсюда и из соотношений (4.22), (4.23), (4.25) в свою очередь следует, что

$$(2t_L - 1)]\, d_D/2\, [\leqslant n - d_D,$$

а поэтому

$$2t_L\, [d_D/2] \leqslant n - (d_D/2).$$

Если сравнить верхние границы для t_1 и t_L, то можно заметить, что оценка для t_L приблизительно в 2 раза меньше оценки для t_1.

Пример 4.10. Рассмотрим (7, 4)-код Хэмминга. Проверочная матрица этого кода была построена в примере 3.5:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Обозначим через s_1, s_2 и s_3 проверочные суммы, коэффициентами которых являются элементы соответственно 1, 2 и 3-й строк матрицы H. Так как

$$s_1 + s_2 = e_0 + e_1 + e_4 + e_6,$$

$$s_1 + s_2 + s_3 = e_0 + e_1 + e_2 + e_5,$$

то проверочные суммы $s_1 + s_2$ и $s_1 + s_2 + s_3$ ортогональны относительно $e_0 + e_1$. Поскольку этот код является циклическим, то двойственный к нему код также циклический. Отсюда, обозначая циклический сдвиг вектора v на одну компоненту вправо через Pv, а сдвиг на i компонент вправо через $P^i v$, получаем

$$P^6(s_1 + s_2) = e_6 + e_0 + e_3 + e_5,$$

$$P^6(s_1 + s_2 + s_3) = e_6 + e_0 + e_1 + e_4.$$

Это означает, что проверочные суммы $P^6(s_1+s_2)$ и $P^6(s_1+s_2+s_2+s_3)$ ортогональны относительно e_0+e_6. Поскольку суммы e_0+e_1 и e_0+e_6 ортогональны в свою очередь относительно e_0 и этот код является циклическим, то, таким образом, мы показали, что рассматриваемый код допускает исправление одиночных ошибок с помощью порогового декодирования в 2 шага и, следовательно, является полностью ортогонализуемым в 2 шага. Этот код представляет собой один из укороченных циклических кодов Рида — Маллера, а описанный здесь метод его декодирования является частным случаем известного общего метода декодирования таких кодов (см. пример 4.12).

Задача 4.9 [26]. Как следует изменить формулировку леммы 4.4, если свойство 2 в определении ортогональности заменить следующим:

2′) для любого $j \notin J$ среди символов $a_{1j}, a_{2j}, \ldots, a_{lj}$ отличными от нуля являются самое большее λ символов.

Краткое решение. Величину $[l/2]$ следует заменить на $[l/2\lambda]$. Доказательство леммы 4.4 в этом случае проводится аналогично.

Методы декодирования, основанные на лемме 4.4, измененной указанным в задаче 4.9 способом, называются методами порогового декодирования с помощью *неортогональных проверок*. Область применения этих методов достаточно широка, но они обладают тем недостатком, что при их использовании значительно возрастает число входов мажоритарных элементов [4]. Обобщению методов мажоритарного декодирования и их модификации посвящен ряд работ [4, 27—31]. Наиболее типичные классы кодов, допускающих мажоритарное декодирование, будут рассмотрены в разд. 4.5.2 и 7.7.2.

4.4. Многочлены Матсона — Соломона

Некоторые способы задания циклических кодов основаны на использовании многочленов *Матсона — Соломона* (МС-многочленов) [3, 4, 32]. Пусть q — число, являющееся степенью некоторого простого числа p, m — целое положительное число, n — целое положительное число, которое делит q^m-1, и β — элемент порядка n поля $GF(q^m)$. Обозначим через V_n векторное пространство размерности n над $GF(q)$. Сопоставим вектору $v=(v_0, v_1, \ldots, v_{n-1}) \in V_n$ многочлен $v(x)=v_0+v_1X+\ldots+v_{n-1}X^{n-1}$. МС-многочлен вектора v, обозначаемый ниже через $MS(v, X)$, определяется следующим образом:

$$MS(v,\ X)=n^{-1}\sum_{j=1}^{n} S_jX^{n-j}, \tag{4.27}$$

где

$$S_j = v(\beta^j). \tag{4.28}$$

Так как $v_l^q = v_l$, то

$$S_j^q = \left(\sum_{i=0}^{n-1} v_i \beta^{ij}\right)^q = \sum_{i=0}^{n-1} v_i \beta^{ijq} = S_{qj} \tag{4.29}$$

(индексы у S_j вычисляются по модулю n). Таким образом,

$$MS(v, \beta^l) = n^{-1} \sum_{j=1}^{n} S_j \beta^{-jl} =$$

$$= n^{-1} \sum_{j=1}^{n} \beta^{-jl} \sum_{i=0}^{n-1} v_i \beta^{ij} = n^{-1} \sum_{i=0}^{n-1} v_i \sum_{j=1}^{n} \beta^{(i-l)j}.$$

Ниже нам понадобится следующая лемма.

Лемма 4.5. *Пусть β — элемент порядка n поля $GF(q^m)$, и пусть i — целое число. Тогда*

1) *если $i \not\equiv 0 \pmod n$, то*

$$n^{-1} \sum_{j=0}^{n-1} \beta^{ij} = 0;$$

2) *если $i \equiv 0 \pmod n$, то*

$$n^{-1} \sum_{j=0}^{n-1} \beta^{ij} = 1.$$

Доказательство. Справедливость части 2 леммы очевидна. Для доказательства части 1 достаточно заметить, что в этом случае $\beta^i - 1 \neq 0$ и

$$(\beta^i - 1) \sum_{j=0}^{n-1} \beta^{ij} = \beta^n - 1 = 0.$$

Из леммы 4.5 и равенства $\sum_{j=1}^{n} \beta^{(j-l)j} = \sum_{j=0}^{n-1} \beta^{(i-l)j}$ следует, что

$$MS(v, \beta^l) = v_l, \quad 0 \leqslant l < n. \tag{4.30}$$

Это равенство показывает, что если среди элементов $\beta^1, \beta^2, \ldots, \beta^n = \beta^0$ имеется r корней $MS(v, X)$, то вес $w(v)$ вектора v равен $n - r$.

Пусть C — q-ичный циклический (n, k)-код с порождающим многочленом

$$g(X) = \prod_{i \in R} (X - \beta^i),$$

где $R \subseteq \{0, 1, \ldots, n-1\}$. Как было показано в разд. 3.6, множество R замкнуто относительно перестановки Π_q: $i \to iq \pmod n$ множества $\{0, 1, \ldots, n-1\}$. Пусть $d_{\text{БЧХ}}$ — минимальное целое число, не принадлежащее R. Если v — произвольный кодовый

вектор кода C, то

$$S_j = v(\beta^j) = 0, \quad j \in R,$$

поскольку $g(X) \mid v(X)$. Следовательно, степень $MS(v, X)$ не больше чем $n - d_{\text{БЧХ}}$. Если v — ненулевой вектор веса $w(v)$, то из формулы (4.30) получаем

$$d_{\text{БЧХ}} \leqslant w(v). \tag{4.31}$$

Таким образом, $d_{\text{БЧХ}}$ является не чем иным, как нижней БЧХ-границей для минимального расстояния.

Обратно, выберем произвольным образом совокупность $(S_1, S_2, \ldots, S_n)$ элементов $GF(q^m)$, удовлетворяющую следующим условиям:

$$\begin{aligned} &S_j = 0, && j \in R, \\ &S_j^q = S_{qj}, && 1 \leqslant j \leqslant n. \end{aligned} \tag{4.32}$$

Пусть

$$a(X) = n^{-1} \sum_{j=1}^{n} S_j X^{n-j},$$

$$v_a = (a(\beta^0), a(\beta^1), \ldots, a(\beta^{n-1})).$$

Тогда из леммы 4.5 для всех $0 \leqslant l < n$ имеем

$$v_a(\beta^l) = \sum_{i=0}^{n-1} a(\beta^i) \beta^{il} = n^{-1} \sum_{i=0}^{n-1} \beta^{il} \sum_{j=1}^{n} S_j \beta^{(n-j)i} =$$

$$= n^{-1} \sum_{j=1}^{n} S_j \sum_{i=0}^{n-1} \beta^{(l-j)i} = S_l,$$

т. е. $v_a \in O$. Отсюда и из равенств (4.27) и (4.28) следует, что

$$MS(v_a, X) = a(X).$$

Пусть $MS(C)$ — множество всех многочленов над $GF(q^m)$ вида $n^{-1} \sum_{j=1}^{n} S_j X^{n-j}$, удовлетворяющих соотношениям (4.32). Из вышеизложенного следует справедливость следующей теоремы.

Теорема 4.4. *Отображение C в $MS(C)$*

$$\varphi\colon\ v \to MS(v, X)$$

и обратное отображение

$$\varphi^{-1}\colon\ a(X) \to v_a = (a(\beta^0), a(\beta^1), \ldots, a(\beta^{n-1}))$$

устанавливают взаимно однозначное соответствие между элементами C и $MS(C)$. Число корней многочлена $MS(v, X)$ в $GF(q^m) - \{0\}$ равно $n - w(v)$.

Это отображение играет в теории кодирования важную роль, поскольку позволяет сводить задачи о весах кодовых векторов к задачам о числе корней многочленов.

Пример 4.11 [32]. Рассмотрим двоичный (23, 13)-код Голея из примера 4.2. В этом случае $n = 23$, $m = 11$, $R = \{1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18\}$ и β — элемент $GF(2^{11})$ порядка 23. Множество $\{0, 1, \ldots, n-1\} - R$ состоит из следующих (2, 23)-циклов: $\{0\}$ и $\{-1 = 22, -2 = 21, -4 = 19, -8 = 15, -16 = 7, -9 = 14, -18 = 5, -13 = 10, -3 = 20, -6 = 17, -12 = 11\}$. В последнем цикле i-е целое число сравнимо с -2^{i-1} по модулю 23 [это связано с тем, что все (2, 23)-циклы — это $\{0\}$, цикл длины 11, содержащий 1, и цикл той же длины, содержащий $-1 = 22$]. Следовательно, полагая $S_{22} = S$, получаем $S_{21} = S^2$, $S_{19} = S^{2^2}$, $S_{15} = S^{2^3}$, $S_7 = S^{2^4}$, $S_{14} = S^{2^5}$, $S_5 = S^{2^6}$, $S_{10} = S^{2^7}$, $S_{20} = S^{2^8}$, $S_{17} = S^{2^9}$, $S_{11} = S^{2^{10}}$. Согласно (4.32), $S_0^2 = S_0$, т. е. $S_0 \in GF(2)$. Как следует из вышеизложенного, для любого кодового вектора

$$MS(v, X) = S_0 + SX + S^2X^2 + S^{2^8}X^2 - S^{2^2}X^4 + S^{2^9}X^6 + S^{2^7}X^8 + \\ + S^{2^5}X^9 + S^{2^{10}}X^{12} + S^{2^7}X^{13} + S^{2^4}X^{16} + S^{2^6}X^{18}, \\ S_0 \in GF(2), \quad S \in GF(2^{11}). \qquad (4.33)$$

Так как $v(\beta^0) = S_0$, то $S_0 = 0$ при четном $w(v)$ и $S_0 = 1$ при нечетном $w(v)$. Ниже будем считать, что $v \neq 0$. Поскольку степень $MS(vX)$ не больше 18, то $w(v)$ не меньше $23 - 18 = 5$. Предположим, что $w(v) = 5$. Тогда $S_0 = 1$ и все корни $MS(v, X)$ — это корни уравнения $X^{23} - 1 = 0$ вида β^i. Произведение этих 18 корней, согласно 4.33, равно S^{-2^6}, так что $(S^{-2^6})^{23} = 1$. Следовательно, $(S^{2^6})^{2^5} = S$ также является корнем $X^{23} - 1$, и показатель S следует вычислять по модулю 23. Равенство (4.33) перепишем в виде

$$MS(v, X) = S_0 + SX + (SX)^2 + (SX)^3 + (SX)^4 + (SX)^6 + (SX)^8 + \\ + (SX)^9 + (SX)^{12} + (SX)^{13} + (SX)^{16} + (SX)^{18}.$$

Тогда, поскольку $MS(v, S^{-1}X)$ — многочлен над $GF(2)$, то, если β^i — корень, β^{2i} — также корень. Так как длина (2, 23)-циклов, отличных от $\{0\}$, равна 11, то число корней $MS(v, S^{-1}X)$ вида β^i не больше 12. В то же время многочлен $MS(v, X)$ также имеет не более 12 корней вида β^i. Получили противоречие. Следовательно, $w(v) \geqslant 6$. Предположим, что $w(v) = 6$. Тогда $S_0 = v(1) = 0$, и все корни многочлена $MS(v, X)X^{-1} = S + S^2X + \ldots + S^{2^6}X^{17}$ должны иметь вид β^i. Произведение этих корней равно S^{1-2^6}. Элемент S^{1-2^6}, а следовательно, и S^{2^6-1} являются корнями уравнения $X^{23} - 1 = 0$. Так как числа $2^6 - 1$ и $2^{11} - 1$ взаимно

простые, то сам элемент S также имеет вид β^i. Вновь получили противоречие. Следовательно, $w(v) \geqslant 7$. Поскольку кодовый вектор v, для которого многочлен $v(X)$ является порождающим многочленом $g(X) = 1 + X + X^5 + X^6 + X^7 + X^9 + X^{11}$, имеет вес 7, то минимальное расстояние (23, 12)-кода Голея равно семи.

Утверждение 4.10. *Рассмотрим код Рида — Соломона, определенный в примере* 4.4. *Множество* $MS(C)$ *представляет собой множество всех многочленов над* $GF(q)$ *степени* $(q-2-r)$ *и менее. Следовательно,* $C = \{(a(\beta^0), a(\beta^1), \ldots, a(\beta^{q-2})) \mid a(X)$ — *многочлены над* $GF(q)$ *степени* $(q-2-r)$ *и менее*$\}$.

Краткое доказательство. Так как $qi \equiv i \pmod{q-1}$, то доказательство непосредственно следует из определений и теоремы 4.4.

4.5. Полиномиальные коды

4. 5. 1. Обобщенные коды Рида — Маллера

Как следует из теоремы 4.4 и утверждения 4.10, код можно задать, указав множество F многочленов, удовлетворяющих определенным условиям, и сопоставив далее каждому многочлену $f \in F$ вектор, компонентами которого являются значения f, получающиеся при подстановке в многочлен f определенных значений переменной. Впервые этот способ использовал Митани [33] для изучения кодов Рида — Маллера [34].

Обозначим через $P_{\nu, m}$ множество всех многочленов степени ν и меньше от m переменных $X_1, X_2, \ldots, X_m$, принимающих значения в поле $GF(q)$[1]). Так как $X^q = X$ для любой переменной X, принимающей значения в $GF(q)$, то любой одночлен из $P_{\nu,m}$ с коэффициентом 1 имеет следующий вид:

$$X_1^{i_1} X_2^{i_2} \ldots X_m^{i_m}, \quad 0 \leqslant i_j < q, \quad 1 \leqslant j \leqslant m.$$

Любой многочлен $f \in P_{\nu,m}$ может быть представлен в виде линейной комбинации таких (нормированных) одночленов с коэффициентами из поля $GF(q)$. Так как различные одночлены линейно независимы (утверждение 2.49), то коэффициенты этой линейной комбинации определяются однозначно. Множество всех линейных комбинаций нормированных одночленов степени ν и меньше с коэффициентами из поля $GF(q^m)$ будем обозначать через $P'_{\nu, m}$.

---

[1]) Коэффициентами этих многочленов также являются элементы $GF(q)$.

Лемма 4.6. *Сумма* $\sum_{a_1 \in GF(q)} \sum_{a_2 \in GF(q)} \dots \sum_{a_m \in GF(q)} a_1^{i_1} a_2^{i_2} \dots a_m^{i_m}$ *равна* $(-1)^m$, *если* $i_1 = i_2 = \dots = i_m = q-1$, *и равна* 0 *во всех остальных случаях.*

Доказательство. Указанная выше сумма равна $\prod_{j=1}^{m} \sum_{a_j \in GF(q)} a_j^{i_j}$.

Положим $s_i = \sum_{a \in GF(q)} a^i$. Если $a \neq 0$, то $s_{q-1} = 1$, так что $a^{q-1} = -1$. Очевидно, $s_0 = 0$. Если $0 < i < q-1$ и β — один из примитивных элементов поля $GF(q)$, то, полагая $a = \beta^j$, из леммы 4.5 получаем

$$s_i = \sum_{j=0}^{q-2} \beta^{ij}.$$

Отсюда следует справедливость леммы.

Пусть α — один из примитивных элементов $GF(q^m)$. Тогда

$$\alpha^i = \sum_{j=0}^{m-1} a_{ij} \alpha^j, \quad 0 \leqslant i \leqslant q^m - 1, \tag{4.34}$$

где $a_{ij} \in GF(q)$. Для простоты значение $f(a_{i0}, a_{i1}, \dots, a_{i,m-1})$ многочлена $f(X_1, \dots, X_m) \in P'_{\nu, m}$ будем обозначать через $f(\alpha^i)$, а значение $f(0, \dots, 0)$ — через $f(0)$. Положим

$$v^e(f) = (f(0), f(\alpha^0), f(\alpha^1), \dots, f(\alpha^{q^m-2})).$$

Вектор, получаемый из $v^e(f)$ удалением первой компоненты, будем обозначать через $v(f)$. Пользуясь леммой 4.6, можно проверить справедливость следующей леммы.

Лемма 4.7. *Пусть* $\nu < m(q-1)$ *и* $f \in P'_{\nu, m}$. *Тогда*

$$\sum_{a \in GF(q^m)} f(a) = 0.$$

Множество $\{v(f) \mid f \in P_{\nu,m}\}$, $0 \leqslant \nu < m(q-1)$, называется *q-ичным обобщенным кодом Рида — Маллера длины* $n = q^m - 1$ *порядка* ν (или сокращенно *ОРМ-кодом*) [3, 4, 34, 35]. Очевидно, что этот код является линейным. Расширение ОРМ-кода назовем *р-ОРМ-кодом*. Как следует из леммы 4.7, р-ОРМ-код — это множество $\{v^e(f) \mid f \in P_{\nu,m}\}$. В случае $q = 2$ р-ОРМ-код называется *кодом Рида — Маллера*, а ОРМ-код — *укороченным циклическим кодом Рида — Маллера*. В случае $m = 1$ ОРМ-код совпадает с *кодом Рида — Соломона* (см. утверждение 4.10). Исследуем основные свойства ОРМ-кодов и р-ОРМ-кодов.

Утверждение 4.11. *Число информационных символов* k_ν *р-ОРМ-кода порядка* ν *и ОРМ-кода порядка* ν *равно общему*

числу совокупностей целых чисел $(i_1, i_2, \ldots, i_m)$, *удовлетворяющих следующим условиям*:

$$0 \leqslant i_j < q,$$

$$\sum_{j=1}^{m} i_j \leqslant \nu.$$

В частности, $k_\nu = \sum_{j=0}^{\nu} \binom{m}{j}$ *при* $q = 2$.

Краткое доказательство. Как указывалось выше, многочлен $f \in P_{\nu,m}$ является линейной комбинацией одночленов $X_1^{i_1} X_2^{i_2} \ldots X_m^{i_m}$, $0 \leqslant i_j \leqslant q$, $1 \leqslant j \leqslant m$, $\sum_{j=1}^{m} i_j \leqslant \nu$. Так как все одночлены линейно независимы, то число информационных символов p-ОРМ-кода порядка ν равно общему числу таких одночленов. Так как $\nu < m(q-1)$, то, согласно лемме 4.7, p-ОРМ-код является расширением ОРМ-кода и имеет то же число информационных символов.

Утверждение 4.12. $k_\nu + k_{m(q-1)-\nu-1} = q^m$.

Краткое доказательство. Согласно утверждению 4.11, $k_{m(q-1)-\nu-1}$ равно числу совокупностей целых чисел $(i_1, i_2, \ldots, i_m)$, удовлетворяющих условиям

$$0 \leqslant i_j < q, \quad 1 \leqslant j \leqslant m,$$

$$\sum_{j=1}^{m} i_j \leqslant m(q-1) - \nu - 1.$$

Полагая $i'_j = q - 1 - i_j$, получим, что это число равно числу совокупностей целых чисел $(i'_1, i'_2, \ldots, i'_m)$, удовлетворяющих условиям

$$0 \leqslant i'_j < q, \quad 1 \leqslant j \leqslant m,$$

$$\sum_{j=1}^{m} i'_j \geqslant \nu + 1.$$

Следовательно, $k_\nu + k_{m(q-1)-\nu-1}$ равно общему числу q^m всех совокупностей $(i_1, \ldots, i_m)$, для которых $0 \leqslant i_j < q$, $1 \leqslant j \leqslant m$.

Теорема 4.5. *Код, двойственный p-ОРМ-коду порядка* ν, *является p-ОРМ-кодом порядка* $m(q-1) - \nu - 1$.

Доказательство. Если $f \in P_{\nu,m}$, $f' \in P_{m(q-1)-\nu-1,\,m}$, то по определению

$$v^e(f)\,[v^e(f')]^T = \sum_{a \in GF(q^m)} f(a) f'(a).$$

Поскольку степень $f \cdot f'$ не превышает $m(q-1)-1$, то, как следует из леммы 4.7, вышеприведенное выражение равно 0. Это означает, что р-ОРМ-код порядка $m(q-1)-\nu-1$ является подкодом кода, двойственного р-ОРМ-коду порядка ν. Из утверждения 4.12 в свою очередь следует, что этот подкод совпадает с двойственным кодом.

Определим *вес* $W_q(h)$ целого числа h, $0 \leqslant h < q^m$, q-ичное разложение которого имеет вид

$$h = \sum_{l=0}^{m-1} h_l q^l, \quad 0 \leqslant h_l < q, \quad 0 \leqslant l < m, \tag{4.35}$$

по формуле

$$W_q(h) = \sum_{l=0}^{m-1} h_l.$$

Теорема 4.6. *ОРМ-код порядка ν является циклическим кодом, множеством корней порождающего многочлена которого является множество* $\{\alpha^h \mid 0 < W_q(h) < m(q-1)-\nu,\ 0 < h < q^m\}$.

Доказательство. Пусть h — произвольное целое число, такое, что $0 < W_q(h) < m(q-1)-\nu$, $0 < h < q^m$, и пусть выражение (4.35) — его q-ичное разложение. Как следует из утверждения 4.11, имеется $k_{m(q-1)-\nu-1}-1$ чисел h, удовлетворяющих указанным выше условиям (—1 появляется из-за того, что число $h = 0$ исключается). Учитывая, что $n = q^m - 1$, из утверждения 4.12 находим, что количество h равно $n - k_\nu$. Следовательно, циклический код C длины n с порождающим многочленом $g(X) = \prod_{\substack{0<h<n \\ 0<W_q(h)<m(q-1)-\nu}} (X - \alpha^h)$ имеет то же число проверочных символов, что и ОРМ-код порядка ν. Таким образом, для завершения доказательства теоремы достаточно показать, что ОРМ-код порядка ν является подкодом кода C. Рассмотрим следующий многочлен f_h от переменных $X_1, X_2, \ldots, X_m$:

$$f_h = \left(\sum_{j=0}^{m-1} \alpha^j X_{j+1}\right)^h.$$

Так как $X_j^q = X_j$, то

$$f_h = \left(\sum_{j=0}^{m-1} \alpha^j X_{j+1}\right)^{\sum_{l=0}^{m-1} h_l q^l} = \prod_{l=0}^{m-1} \left(\sum_{j=0}^{m-1} \alpha^{jq^l} X_{j+1}\right)^{h_l}.$$

Степень f_h не превышает $W_q(h)$ и, следовательно, не больше $m(q-1)-\nu-1$. Из формулы (4.34) имеем

$$f_h(\alpha^i) = \left(\sum_{j=0}^{m-1} a_{ij} \alpha^j\right)^h = \alpha^{ih}.$$

Отсюда следует, что $v(f_h)=(\alpha^0, \alpha^h, \alpha^{2h}, \ldots, \alpha^{(n-1)h})$. Если f — произвольный вектор из $P_{\nu,m}$ и $v(f)=(f(\alpha^0), f(\alpha^1), \ldots, f(\alpha^{n-1}))=(v_0, v_1, \ldots, v_{n-1})$, то

$$\sum_{l=0}^{n-1} v_i \alpha^{ih} = \sum_{\substack{a \in GF(q^m) \\ a \neq 0}} f(a) f_h(a).$$

Так как степень $f \cdot f_h$ меньше $m(q-1)$ и $f_h(0)=f_h(0, \ldots, 0)=0$, то, согласно лемме 4.7, сумма в правой части последнего равенства равна 0. Отсюда и из формулы (3.24) следует, что $v(f) \in C$. Таким образом, ОРМ-код порядка ν является подкодом кода C, так что они совпадают.

Теорема 4.7. *Пусть* $m(q-1)-\nu = Q(q-1)+R$, $0 \leqslant Q < m$, $0 \leqslant R < q-1$. *Тогда ОРМ-код порядка* ν *имеет минимальное расстояние*

$$d_{\text{мин}} = (R+1)q^Q - 1$$

и является подкодом БЧХ-кода с конструктивным расстоянием $d_{\text{мин}}$ [1]) *БЧХ-кода с* $l=0$).

Доказательство. Так как $\{h \mid 0 < W_q(h) < Q(q-1)+R,\ 0 < h < q^m\} \supseteq \left\{h/h = \sum_{l=0}^{Q-1} h_l q^l + h_Q q^Q,\ 0 \leqslant h_l < q\ (0 \leqslant l < Q),\ 0 \leqslant h_Q < R,\ h \neq 0\right\}$, то порождающий многочлен ОРМ-кода порядка ν имеет среди своих корней элементы $\alpha^1, \alpha^2, \ldots, \alpha^{(R+1)q^Q-2}$. Это означает, что ОРМ-код порядка ν является подкодом БЧХ-кода с конструктивным расстоянием $(R+1)q^Q-1$. Из нижней БЧХ-границы получаем

$$d_{\text{мин}} \geqslant (R+1)q^Q - 1. \tag{4.36}$$

Пусть β — один из примитивных элементов $GF(q)$. Положим

$$f_{\text{мин}} = (X_1^{q-1}-1)(X_2^{q-1}-1)\ldots(X_{m-Q-1}^{q-1}-1)\sum_{i=0}^{q-2-R}(X_{m-Q}-\beta^i).$$

Так как $(q-1)(m-Q-1)+q-1-R=\nu$, то необходимым условием того, что $f_{\text{мин}} \in P_{\nu,m}$, $f_{\text{мин}} \neq 0$, является выполнение равенств $X_1 = X_2 = \ldots = X_{m-Q-1} = 0$. Поскольку $X_{m,Q} \neq \beta^i$ $(0 \leqslant i \leqslant q-2-R)$, число совокупностей значений $X_1, \ldots, X_m$, для которых $f_{\text{мин}} \neq 0$, равно $(R+1)q^Q$. Учитывая, что в число этих совокупностей входит и $(0, 0, \ldots, 0)$, получаем, что вес $v(f_{\text{мин}})$ равен $(R+1)q^Q-1$. Таким образом, в формуле (4.36) в действительности имеет место знак равенства.

---

[1]) БЧХ-код с $l=0$.

Пусть U — подмножество $GF(q^m)$. Для каждого $0 \leqslant i < q^m - 1$ определим величину u_i, положив $u_i = 1$, если $\alpha^i \in U$, и $u_i = 0$, если $\alpha^i \notin U$; положим $u_\infty = 1$, если $0 \in U$, и $u_\infty = 0$, если $0 \notin U$. Вектор $u^e = (u_\infty, u_0, u_1, \ldots, u_{q^m-2})$ называется характеристическим вектором подмножества U (вектор $u = (u_0, u_1, \ldots, u_{q^m-2})$, получающийся из u^e удалением первой компоненты, называется укороченным характеристическим вектором).

Теорема 4.8. *Пусть U — аффинное подпространство*[1]) *над $GF(q)$ пространства $GF(q^m)$ размерности r. Укороченный характеристический вектор u и характеристический вектор u^e подпространства U являются кодовыми векторами соответственно ОРМ-кода и p-ОРМ-кода порядка $(m - r)(q - 1)$.*

Доказательство. Из разд. 2.6.3 следует существование $(m-r)$ строк $b^{(1)}, b^{(2)}, \ldots, b^{(m-r)}$ длины m с компонентами из $GF(q)$ и $m - r$ элементов $c_1, c_2, \ldots, c_{m-r}$ поля $GF(q)$, таких, что

$$U = \left\{ \sum_{l=1}^{m} \alpha^{l-1} X_l \mid b^{(i)} X^T = c_i, \quad 1 \leqslant i \leqslant m - r \right\},$$

где $X = (X_1, X_2, \ldots, X_m)$. Положим

$$f_U = \prod_{i=1}^{m-r} \left(1 - (b^{(i)} X^T - c_i)^{q-1}\right).$$

Тогда выражение $(b^{(i)} X^T - c_i)^{q-1}$ равно 0, если $b^{(i)} X^T = c_i$, и равно 1 во всех остальных случаях, так что $v(f_U) = u$. Тогда, очевидно, $v^e(f_U) = u^e$.

Утверждение 4.13. *Четный подкод укороченного циклического кода Рида — Маллера 1-го порядка является кодом максимальной длины. Укороченный циклический код Рида — Маллера порядка $(m - 2)$ представляет собой код Хэмминга.*

Это утверждение непосредственно следует из теоремы 4.6.

Утверждение 4.14. *Минимальное расстояние q-ичного БЧХ-кода длины $n = q^m - 1$ с конструктивным расстоянием $(R + 1)q^Q - 1$ (где $0 \leqslant Q < m$, $0 \leqslant R < q - 1$) и $l = 0$ совпадает с конструктивным расстоянием.*

Краткое доказательство. Справедливость данного утверждения непосредственно следует из доказательства теоремы 4.7.

Выше были описаны ОРМ-коды длины $n = q^m - 1$, но могут быть построены и ОРМ-коды длины $n_1 = (q^m - 1)/b$, где b — произвольный делитель $q - 1$.

---

[1]) См. разд. 2.6.3.

Утверждение 4.15. *Пусть* $f = X_1^{i_1} X_2^{i_2} \ldots X_m^{i_m}$ *и* $b \Big| \sum_{j=1}^{m} i_j$. *Тогда вектор* $v(f)$ *получается* b*-кратным повторением первых* $(q^m-1)/b$ *своих компонент.*

Краткое доказательство. Так как $\alpha^{(q-1)n_1} = \alpha^{(q^m-1)(q-1)/b} = 1$, то $\alpha^{n_1} \in GF(q)$. Из формулы (4.34) для произвольных $0 \leqslant i < n_1$ и $0 \leqslant l < b$ получаем

$$\alpha^{i+ln_1} = \alpha^{ln_1} \sum_{j=0}^{m-1} a_{ij}\alpha^j = \sum_{j=0}^{m-1} a_{i+ln_1,\ j}\alpha^j.$$

Так как $\alpha^{ln_1} \in GF(q)$, то $a_{i+ln_1,\ j} = \alpha^{ln_1} a_{ij}$, $0 \leqslant j < m$. Следовательно,

$$f(\alpha^{i+ln_1}) = \alpha^{ln_1 \sum_{j=0}^{m-1} i_j} \prod_{j=0}^{m-1} a_{ij}^{i_j} = f(\alpha^i).$$

Пусть $P_{\nu,\, m,\, b}$ — множество всех линейных комбинаций одночленов степени ν и меньше с коэффициентами из $GF(q)$, удовлетворяющих условиям утверждения 4.15. Тогда для произвольного $f \in P_{\nu,\, m,\, b}$ вектор $v(f)$ получается b-кратным повторением первых n_1 своих компонент. Если $v'(f)$ — вектор длины n_1, состоящий из первых n_1 компонент $v(f)$, то множество $\{v'(f) \mid f \in P_{\nu b,\, m,\, b}\}$ называют ОРМ-кодом длины n_1 порядка ν.

Утверждение 4.16. *ОРМ-код длины* n_1 *порядка* ν *является циклическим кодом, множеством корней порождающего многочлена которого является множество* $\{\alpha^h \mid 0 < W_q(h) < m(q-1) - \nu b,\ 0 < h < q^m,\ b \mid W_q(h)\}$.

Краткое доказательство. Утверждения 4.11 и 4.12 можно легко обобщить. Тогда, если заметить, что для любых многочленов

$$f \in P_{b\nu,\, m,\, b}, \quad f' \in P_{m(q-1)-(\nu+1)b,\, m,\, b}$$

имеет место равенство

$$bv'(f)[v'(f')]^T = v(f)[v(f')]^T = 0,$$

то справедливость данного утверждения будет следовать непосредственно из доказательства теоремы 4.6.

Утверждение 4.17. *Теорема 4.7 остается справедливой, если длину кода заменить на* n_1, ν *— на* νb *и* $d_{\text{мин}}$ *— на* $[(R+1)q^Q - 1]/b$.

Идея доказательства. Воспользоваться утверждением 4.16, положить $\beta = \alpha^b$ и далее воспользоваться нижней БЧХ-границей.

Рассмотрим частный случай, когда $b = q - 1$. При этом $n_1 = (q^m - 1)/(q - 1)$. Для любого $0 \leqslant i < n_1$ множество

$\{(a_{l+ln_1,\,0}, a_{l+ln_1,\,1}, \ldots, a_{l+ln_1,\,m-1}) \mid 0 \leqslant l < q-1\}$ из $q-1$ векторов представляет собой некоторое множество совпадающих точек проективной геометрии $PG(m, q)$[1]). С помощью отображения, задаваемого формулой (4.34), этим векторам могут быть сопоставлены элементы α^{i+ln_1}, $0 \leqslant l < q-1$. С помощью такого отображения устанавливается взаимно однозначное соответствие между точками $PG(m, q)$ и множествами $\{\alpha^i \mid 0 \leqslant i < n_1\}$.

Ниже точки $PG(m, q)$ будем представлять с помощью соответствующих элементов α^i. Пусть U — проективное подпространство размерности r проективного пространства $PG(m, q)$. Для каждого $0 \leqslant i < n_1$ определим величину u_i следующим образом: $u_i = 1$, если $\alpha^i \in U$, и $u_i = 0$, если $\alpha^i \notin U$. Назовем вектор $u = (u_0, u_1, \ldots, u_{n_1-1})$ характеристическим вектором U.

Утверждение 4.18. *Пусть U — проективное подпространство размерности r проективного пространства $PG(m, q)$. Характеристический вектор подпространства U является кодовым вектором ОРМ-кода длины $n_1 = (q^m - 1)/(q-1)$ порядка $m-r-1$.*

Краткое доказательство. Согласно определению проективного подпространства размерности r, существует такое подпространство $\bar{U}$ размерности $r+1$, что $U = \{\alpha^i \mid \alpha^i \in \bar{U},\ 0 \leqslant i < n_1\}$. В свою очередь в $\bar{U}$ существуют такие $(m-r-1)$ вектор-строк $(b^{(1)}, b^{(2)}, \ldots, b^{(m-r-1)})$ размерности m с компонентами из $GF(q)$, что (см. разд. 2.6.3)

$$\bar{U} = \left\{ \sum_{l=0}^{m} \alpha^{l-1} X_l \mid b^{(i)} X^T = 0, \quad 1 \leqslant i < m-r \right\},$$

где $X = (X_1, X_2, \ldots, X_m)$. Положим

$$f_U = \prod_{i=1}^{m-r-1} (1 - (b^{(i)} X^T)^{q-1}).$$

Тогда $f_U \in P_{(m-r-1)(q-1),\, m,\, q-1}$ и $v'(f_U)$ совпадает с характеристическим вектором подпространства U.

4.5.2. Полиномиальные коды и двойственные к ним коды

Множество C_0 всех кодовых слов кода C над $GF(q^s)$, все компоненты которых являются элементами подполя $GF(q)$, называется *$GF(q)$-подкодом кода C* или *подкодом над подполем*. Если C — линейный код, то C_0 также является линейным кодом; если C — циклический код, то циклическим является также и код C_0.

Теорема 4.9. *Пусть β — элемент порядка n поля $GF(q^{sm})$, и пусть C — циклический код длины n над $GF(q^s)$ с порождающим*

[1]) См. разд. 2.6.3.

многочленом $g(X) = \prod_{i \in R} (X - \beta^i)$. *Тогда порождающий многочлен* $GF(q)$*-подкода* C_0 *кода* C *имеет следующий вид:*

$$g_0(X) = \prod_{i \in R_q} (X - \beta^i),$$

где R_q *— минимальное множество, содержащее* R *и замкнутое относительно перестановки* Π_q: $i \to iq \pmod n$.

Доказательство. Если v — произвольный вектор над $GF(q)$, принадлежащий C, то $v(\beta^i) = 0$, $i \in R$, так что $v^q(\beta^i) = v(\beta^{iq}) = 0$. Следовательно, $g_0(X) \mid v(X)$.

Обратно, пусть v — произвольный вектор длины n над $GF(q)$, и пусть $g_0(X) \mid v(X)$. Тогда, так как $g(X) \mid g_0(X)$, то $g(X) \mid v(X)$, а следовательно, $v \in C$.

Например, БЧХ-коды с $l = 0$ являются подкодами над подполем кода Рида — Соломона.

Подкоды над подполем ОРМ-кодов [1]) называются *полиномиальными кодами* [37]. БЧХ-коды при $l = 0$ и укороченные циклические коды Рида — Маллера являются полиномиальными кодами. Кроме того, как будет показано ниже, евклидово-геометрические коды [27, 35] и проективно-геометрические коды [27, 39, 40] являются кодами, двойственными к полиномиальным кодам [4, 37, 41]. Основные свойства полиномиальных кодов можно вывести из теорем 4.6—4.9 [4, 37].

Утверждение 4.19. *Множеством корней порождающего многочлена* $GF(q)$*-подкода* q^s*-ичного ОРМ-кода длины* $(q^{ms} - 1)/b$ (b *— делитель* $q^s - 1$) *порядка* ν *является множество* $\{\alpha^h \mid 0 < h < \min_{0 \leqslant l \leqslant s} W_{q^s}(hq^l) < m(q^s - 1) - \nu b,\ 0 < h < q^m,\ b \mid W_{q^s}(h)\}$.

Идея доказательства. Это утверждение легко доказать, используя утверждение 4.16 и теорему 4.9.

Утверждение 4.20. *Пусть* $m(q^s - 1) - \nu = Q(q^s - 1) + R$, $0 \leqslant Q < m$, $0 \leqslant R < q - 1$, *и пусть минимальный вес* q*-ичного БЧХ-кода* C_0 *длины* q^{s-1} *с конструктивным расстоянием* $R + 1$ *и* $l = 0$ *совпадает с его конструктивным расстоянием. Тогда*

1) *минимальный вес* $d_{\text{мин}}$ $GF(q)$*-подкода* q^s*-ичного ОРМ-кода длины* $q^{ms-1} - 1$ *порядка* ν *равен* $(R + 1)q^{sQ} - 1$;

2) *минимальный вес* q*-ичного БЧХ-кода длины* $q^{ms} - 1$ *с конструктивным расстоянием* $(R + 1)q^{sQ} - 1$ *и* $l = 0$ *равен его конструктивному расстоянию.*

Идея доказательства. Доказательство теоремы 4.7, за исключением того, что вектор $v(f_{\text{мин}})$ должен быть вектором над

[1]) В общем случае длина кода равна $(q^m - 1)/b$, где b — делитель $q - 1$.

$GF(q)$, полностью сохраняется и для подкода над подполем. Пусть $f(X)$ — МС-многочлен кодового вектора минимального веса из C_0. Степень $f(X)$ по определению меньше чем $q^s - 2 - R$. Пусть β — один из примитивных элементов поля $GF(q^s)$. Тогда, так как вес вектора $(f(\beta^0), f(\beta^1), \ldots, f(\beta^{q^s-2}))$ равен $R+1$, то многочлен $f(X)$ имеет степень $q^s - 2 - R$ и ровно $q^s - 2 - R$ корней в поле $GF(q^s)$. Если положить

$$f_{\text{мин}} = \prod_{i=1}^{m-Q-1} (X_i^{q-1} - 1) f(X_{m-Q}),$$

то $v(f_{\text{мин}})$ будет вектором над $GF(q)$. Доказательство части 2 утверждения 4.20 аналогично доказательству утверждения 4.14.

Пример 4.12. Пусть p — простое число и $q = p^s$. Код, двойственный $GF(p)$-подкоду q-ичного р-ОРМ кода длины $n = q^m$ и порядка $[(m - r - 1)(q - 1)]$, называется *расширенным евклидово-геометрическим кодом размерности* (r, s) (сокращенно р-ЕГ-кодом). Эти коды являются расширениями евклидово-геометрических кодов (ЕГ-кодов), введенных Уэлдоном [38]. Как следует из теоремы 4.5, в случае $q = 2$ и $s = 1$ р-ЕГ-код является кодом Рида — Маллера порядка r. Поле $GF(q^m)$ является векторным пространством размерности m над $GF(q)$, но характеристический вектор произвольного аффинного подпространства размерности $(r+1)$ этого пространства, согласно теореме 4.8, принадлежит р-ОРМ-коду порядка $[(m - r - 1)(q - 1)]$. Так как вектор u является вектором над $GF(q)$, то он также принадлежит подкоду над подполем. Рассмотрим произвольное аффинное подпространство U размерности r над $GF(q)$ пространства $GF(q^m)$. Из утверждения 2.76 следует, что число аффинных подпространств размерности $r+1$, содержащих U и не имеющих других общих точек, кроме точек U, равно $(q^{m-r} - 1)/(q - 1)$. Пусть $u^{(1)}, u^{(2)}, \ldots, u^{(d_r)}$, $d_r = (q^{m-r} - 1)/(q - 1)$ — характеристические векторы этих подпространств. Проверочные суммы, коэффициентами которых являются компоненты этих характеристических векторов, ортогональны относительно линейной e-суммы, коэффициентами которой являются компоненты характеристического вектора аффинного подпространства U. Аналогичными свойствами обладают и аффинные подпространства меньшей размерности. Так как $d_r < d_{r-1} < \ldots$, то р-ЕГ-код размерности (r, s) допускает исправление ошибок веса до $[(q^{m-r} - 1)/2(q - 1)]$ с помощью порогового декодирования за $(r+1)$ шагов.

Утверждение 4.21. *Множеством корней порождающего многочлена р-ичного ЕГ-кода длины* $n = p^{ms} - 1$ *и размерности*

(r, s) является множество

$$\{\alpha^l \mid 0 < \max_{0 \leqslant l < s} W_{p^s}(ip^l) \leqslant (m-r-1)(p^s-1),\ 0 < i < p^{ms}\}.$$

Краткое доказательство. Как следует из примера 3.1, ЕГ-код размерности (r, s) является кодом, двойственным четному подкоду C_e подкода над подполем ОРМ-кода размерности $[(m-r-1)(p^s-1)]$. Множеством корней порождающего многочлена кода C, согласно утверждению 4.19, является множество

$$\{\alpha^h \mid 0 \leqslant \min_{0 \leqslant l < s} W_{p^s}(hp^l) < (r+1)(p^s-1),\quad 0 \leqslant h < p^{ms}\}.$$

Чтобы элемент α^i был корнем порождающего многочлена двойственного кода, необходимо, чтобы элемент $\alpha^{-i} = \alpha^{p^{ms}-i-1}$ не принадлежал этому множеству; соответствующее условие на i заключается в том, что $m(p^s-1) > \min\limits_{0 \leqslant l < s} W_{p^s}((p^{ms}-1-i)p^l) \geqslant$ $\geqslant (r+1)(p^s-1)$. Кроме того, заметим, что $W_{p^s}((p^{ms}-1-i)p^l) = m(p^s-1) - W_{p^s}(ip^l)$. Отсюда следует справедливость утверждения 4.21.

Утверждение 4.22. *ЕГ-код может быть декодирован точно так же, как и р-ЕГ-код.*

Идея доказательства. В примере 4.12 следует рассмотреть только такие аффинные подпространства $GF(q^m)$, которые не содержат 0. При этом ошибки веса до $[(d_r-1)/2]$ могут быть исправлены с помощью порогового декодирования за $(r+1)$ шагов.

По определению р-ЕГ-кода двойственный к нему код содержит характеристические векторы всех аффинных подпространств размерности $r+1$, однако совсем не очевидно, что он может быть натянут только на эти векторы. В действительности же это так [42, 43]. Это означает, что из всех кодов, допускающих пороговое декодирование, основанное на описанных выше свойствах аффинных подпространств, ЕГ-коды имеют максимальное число информационных символов.

Утверждение 4.23. *Примитивный БЧХ-код с параметрами $n = 15$, $l = 0$ и $t = 3$ из примера 4.1 является укороченным циклическим кодом Рида — Маллера первого порядка и допускает исправление ошибок веса 3 и менее с помощью порогового декодирования в 2 шага.*

Краткое доказательство. Корнями порождающего многочлена этого кода являются элементы множества $\{\alpha^i \mid i = 2^0, 2^1, 2^2, 2^3, 2^0+2^1, 2^1+2^2, 2^2+2^3, 2^0+2^3, 2^0+2^2, 2^1+2^3\}$. С другой стороны, как следует из примера 4.12, укороченный циклический код Рида — Маллера длины 15 первого порядка является двоич-

ным ЕГ-кодом размерности (1, 1). Полагая в утверждении 4.21 $m = 4$, $p = 2$ и $s = 1$, получаем, что корнями порождающего многочлена этого кода являются элементы множества $\{\alpha^i \mid 0 < W(i) \leqslant 2\}$. Следовательно, оба кода совпадают и, согласно примеру 4.12, допускают исправление ошибок веса 3 и менее с помощью порогового декодирования в 2 шага.

Пример 4.13. Пусть p — простое число и $q = p^s$. Код, двойственный $GF(p)$-подкоду C_0 q-ичного ОРМ-кода C длины $n_1 = (q^m - 1)/(q - 1)$ и порядка $(m - r - 1)$, называется *проективно-геометрическим кодом размерности* (r, s) (или сокращенно ПГ-кодом) [1]). Согласно утверждению 4.18, характеристические векторы u проективных подпространств размерности r проективного пространства $PG(m, q)$ принадлежат C. Так как u — вектор над $GF(p)$, то он принадлежит также и C_0. Рассмотрим произвольное проективное подпространство U размерности $r - 1$ проективного пространства $PG(m, q)$. Существует $d_r = (q^{m-r} - 1)/(q - 1)$ проектных подпространств размерности r, содержащих U и не имеющих других общих точек, кроме точек из U. Проверочные суммы, коэффициентами которых являются компоненты этих характеристических векторов, ортогональны относительно линейной e-суммы, коэффициентами которой являются компоненты характеристического вектора подпространства U. Аналогичными свойствами обладают и проектные подпространства меньшей размерности. Это означает, что ПГ-код размерности (r, s) допускает исправление ошибок веса до $[(q^{m-r} - 1)/2(q - 1)]$ с помощью порогового декодирования в r шагов.

В случае $p = 2$ и $s = 1$ проективно-геометрические коды являются четными подкодами укороченных циклических кодов Рида — Маллера.

Утверждение 4.24. *Корнями порождающего многочлена p-ичного проективно-геометрического кода размерности (r, s) и длины $n_1 = (p^{ms} - 1)/(p^s - 1)$ являются элементы множества*

$$\{\alpha^i \mid 0 \leqslant \max_{0 \leqslant l < s} W_{p^s}(ip^l) \leqslant (m - r - 1)(p^s - 1),\ 0 < i < p^{ms},\ (p^s - 1) \mid W_{p^s}(i)\},$$

где β — элемент $GF(p^{ms})$ порядка n_1.

Идея доказательства. Вывод этого утверждения из утверждения 4.16 почти аналогичен доказательству утверждения 4.21.

Утверждение 4.25. *Код максимальной длины с блоковой длиной $2^m - 1$ является двоичным проективно-геометрическим кодом размерности $(1, 1)$ и, следовательно, допускает исправление*

[1]) Иногда параметры ПГ-кода вводятся несколько отличным способом.

ошибок веса до $2^{m-2}-1$ *с помощью порогового декодирования в один шаг. Код из примера 4.9 является одним из таких кодов.*

Идея доказательства. Следует воспользоваться примером 4.13 и утверждением 4.24.

Циклические разностные коды [44] также являются кодами, двойственными полиномиальным кодам [37]. Изучены расширения кодов, основанных на конечных геометриях [45, 46], и методы порогового декодирования кодов, двойственных произвольным полиномиальным кодам [43]. Нахождение числа информационных символов полиномиальных кодов в общем случае является трудной задачей, а полученные формулы являются очень сложными [47, 48].

Задача 4.26. Пусть $f(X_1, X_2, \ldots, X_m) \in P_{v,m}$. Найти для $v(f)$ многочлен Матсона — Соломона $MS(v(f), X)$.

Краткое решение. По определению i-я компонента вектора $v(f)$ равна $f(a_{i0}, a_{i1}, \ldots, a_{i,m-1})$, где $\alpha^i = \sum_{j=0}^{m-1} a_{ij}\alpha^j$. С другой стороны, из свойств $MS(v(f), X)$ и формулы (4.30) имеем

$$MS(v(f), \alpha^i) = f(a_{i0}, a_{i1}, \ldots, a_{i,m-1}), \quad 0 \leqslant i < q^m - 1. \quad (4.37)$$

Пусть $\gamma_0, \gamma_1, \ldots, \gamma_{m-1}$ — базис, двойственный базису $1, \alpha, \alpha^2, \ldots, \alpha^{m-1}$ (см. утверждение 2.81). Тогда

$$T_q(\alpha^i\gamma_i) = \begin{cases} 0, & i \neq j, \\ 1, & i = j, \end{cases} \quad (4.38)$$

где $0 \leqslant j < m$, $0 \leqslant i < m$. Если положить $\phi(X) = f(T_q(\gamma_0 X), T_q(\gamma_1 X), \ldots, T_q(\gamma_{m-1}X))$, то из формул (4.34) и (4.38) будет следовать, что

$$\phi(\alpha^i) = f(a_{i0}, a_{i1}, \ldots, a_{i,m-1}), \quad 0 \leqslant i < q^m - 1.$$

Многочлен, удовлетворяющий равенству (4.37), определяется однозначно (теорема 4.4), так что $\phi(X) = MS(v(f), X)$.

4.6. Каскадные коды и коды Юстесена

4.6.1. Каскадные коды [1])

Пусть C_O — линейный (n_O, k_O)-код над $GF(q^{k_I})$ и C_I — линейный (n_I, k_I)-код над $GF(q)$. *Каскадным кодом* [49] с внешним кодом C_O и внутренним кодом C_I называется линейный q-ичный

[1]) Каскадным кодам посвящена книга Э. Л. Блоха и В. В. Зяблова [72]. — *Прим. ред.*

$(n_O n_I, k_O k_I)$-код, который строится следующим образом. Вектор v длины $n_O n_I$ в данном разделе будет удобно представлять в виде матрицы размера $n_O \times n_I$ с элементами $v_{ij} (0 \leqslant i < n_O, 0 \leqslant j < n_I)$; элемент v_{ij} этой матрицы будем также называть (i, j)-компонентой вектора v.

1) Информационные компоненты кода C представляют собой вектор длины $k_O k_I$, который мы также представим в виде матрицы размера $k_O \times k_I$ (каждую строку этой матрицы можно рассматривать как представление некоторого элемента поля $GF(q^{k_I})$ в виде вектора над $GF(q)$, пусть v_i — элемент $GF(q^{k_I})$, представлением которого является i-я строка).

2) Пусть $(v_0, v_1, \ldots, v_{n_O-1})$ — кодовый вектор кода C_O, информационные компоненты которого совпадают с $v_0, v_1, \ldots, v_{k_O-1}$ (по определению существует ровно один такой вектор).

3) Пусть ψ — взаимно однозначное отображение [1]) $GF(q^{k_I})$ в C_I (например, каждому элементу из $GF(q^{k_I})$ можно взаимно однозначно сопоставить кодовый вектор кода C_I, информационные компоненты которого представляют собой k_I компонент вектора, являющегося представлением этого элемента в виде вектора над $GF(q)$). Матрицу размера $n_O \times n_I$, i-й строкой которой является $\psi(v_i)$, $0 \leqslant i < n_O$, будем считать кодовым вектором кода C.

Пусть d_O и d_I — соответственно минимальные веса кодов C_O и C_I. Как видно из описания построения каскадного кода, вес вектора $(v_0, v_1, \ldots, v_{n_O-1}) \in C_O$, не все компоненты $v_0, v_1, \ldots, v_{k_O-1}$ которого равны нулю, не меньше d_O, а следовательно, не меньше d_O строк являются ненулевыми строками и их вес в свою очередь не меньше d_I. Таким образом, минимальный вес каскадного кода не меньше чем $d_O d_I$.

Используя в качестве внешнего кода код Рида — Соломона и подбирая специальным образом внутренний код, Форни показал, что в двоичном симметричном канале при любых скоростях, меньших пропускной способности, вероятность ошибочного декодирования для каскадного кода с ростом n_O стремится экспоненциально к нулю [49].

Все строки кодового слова описанного выше каскадного кода являются кодовыми словами одного и того же кода C_I. Можно, однако, рассматривать также каскадные коды, в которых для каждой строки используется свой линейный (n_I, k_I)-код C_{Ii}. Коды Юстесена являются кодами такого типа [50].

---

[1]) Поле $GF(q^{k_I})$ здесь рассматривается как линейное пространство над $GF(q)$ размерности k_I.

Что касается декодирования каскадных кодов, то сначала проводится декодирование каждой принятой строки $(v'_{i0}, v'_{i1}, \ldots, v'_{i, n_I-1})$; при этом используются методы декодирования кода C_I. Информационные части векторов, полученных в результате этого декодирования (первые k_I компонент), можно рассматривать как элементы $v_i^{(d)}$ поля $GF(q^{k_I})$. Далее, используя методы декодирования кода C_O, следует провести декодирование вектора $(v_0^{(d)}, v_1^{(d)}, \ldots, v_{n_O-1}^{(d)})$ [49].

Пример 4.14 [69]. Пусть n — положительное нечетное число, m — минимальное целое число, такое, что $n \mid 2^m - 1$, γ — элемент $GF(2^m)$ порядка n и C_I — двоичный циклический код длины n, проверочным многочленом которого является минимальный многочлен $h_I(X)$ элемента γ. Поскольку корни порождающего многочлена представляют собой элементы множества $\{\gamma^0, \gamma^1, \ldots, \gamma^{n-1}\} = \{\gamma^1, \gamma^2, \gamma^{2^2}, \ldots, \gamma^{2^{m-1}}\}$, то многочлен $MS(v, X)$, соответствующий кодовому вектору v кода C_I, имеет вид

$$MS(v, X) = S_1X^{n-1} + (S_1X^{n-1})^2 + (S_1X^{n-1})^{2^2} + \ldots$$

$$\ldots + (S_1X^{n-1})^{2^{m-1}} = T_2(S_1X^{-1}), \quad S_1 \in GF(2^m). \qquad (4.39)$$

Согласно теореме 4.4, отображение ψ: $S_1 \to MS(v, X) \to v$ является взаимно однозначным линейным отображением m-мерного линейного пространства $GF(2^m)$ над полем $GF(2)$ в C_I. В качестве C_O возьмем 2^m-ичный БЧХ-код длины $n_O = 2^m + 1$ с k_O информационными символами из примера 4.5 или утверждения 4.3. Пусть $MS(C_O)$ — множество всех МС-многочленов кодовых векторов кода C_O и β — элемент $GF(2^{2m})$ порядка $2^m + 1$ (поскольку $2^m + 1 \mid 2^{2m} - 1$, то такой элемент существует). Обозначим через C каскадный код с внешним кодом C_O и внутренним кодом C_I (получаемый с помощью отображения ψ). Код C является q-ичным линейным (n_On, k_Om)-кодом. Если фиксировать один из кодовых векторов $v^{(O)}$ кода C_O, то, согласно определению кода C, фиксируется кодовый вектор $\psi(v^{(O)})$[1]), соответствующий $v^{(O)}$. Элементу $\phi \in MS(C_O)$ сопоставим вектор $v^{(O)}(\phi) = (\phi(\beta^0), \phi(\beta^1), \ldots, \phi(\beta^{n_O-1}))$ из C_O. При этом (i, j)-компонента кодового вектора $v^{(O)}(\phi)$ (для простоты обозначаемого ниже через $v(\phi)$), согласно формуле (4.39), равна

$$T_2(\phi(\beta^i)\gamma^{-j}). \qquad (4.40)$$

Если использовать $\gamma^{-1}\phi \in MS(C_O)$ вместо ϕ, то (i, j)-компо-

[1]) Кодовый вектор, который получается путем замены каждой компоненты $v_i^{(O)}$ вектора $v^{(O)}$ на $\psi(v_i^{(O)})$.

нента $v(\gamma^{-1}\phi)$ будет равна

$$T_2(\phi(\beta^i)\gamma^{-i-1}).$$

Так как $\gamma^n = 1$, то вектор $v(\gamma^{-1}\phi)$ может быть получен с помощью однократного циклического сдвига каждой строки $v(\phi)$. Аналогично, рассматривая $\phi'(X) = \phi(\beta X) \in MS(C_0)$, можно убедиться в том, что $v(\phi')$ является вектором, получающимся однократным циклическим сдвигом каждого столбца $v(\phi)$. Следовательно, код C инвариантен относительно циклических перестановок строк и столбцов. Так как n — нечетное число и $n \mid 2^m - 1$, то $(n, 2^m + 1) = 1$. Если положить $\alpha = \beta\gamma$, то порядок элемента α будет равен $n(2^m+1)$. Пусть $I(h)$ и $J(h)$ — остатки от деления h, $0 \leqslant h < n(2^m + 1)$, соответственно на $2^m + 1$ и n. Тогда $\alpha^h = \beta^h\gamma^h = \beta^{I(h)}\gamma^{J(h)}$. Отображение, сопоставляющее h пару чисел $(I(h), J(h))$, переводит двумерные матрицы в одномерные векторы. Если П — перестановка, представляющая собой последовательность однократных циклических перестановок строк или столбцов, то при этой перестановке $h \to h + 1$, и это соответствует циклической перестановке одномерного вектора. Следовательно, при таком расположении компонент код C оказывается циклическим. Так как, согласно примеру 4.5, минимальный вес C_0 равен 2^m+2-k_0, то минимальный вес C равен $(2^m+2-k_0)d_1$, где d_1 — минимальный вес C_1. Необходимым условием того, что α^h представляет собой корень проверочного многочлена кода, является наличие в C хотя бы одного кодового слова, МС-многочлен которого содержит X^{n-h} с ненулевым коэффициентом. Пусть a — элемент, обратный элементу n по модулю $2^m + 1$, и пусть b — элемент, обратный элементу $2^m + 1$ по модулю n. Так как $a = \beta\gamma$, то

$$\beta = \alpha^{an},$$

$$\gamma = \alpha^{b(2^m+1)}.$$

Следовательно, из определений $I(h)$ и $J(h)$ имеем

$$\beta^{I(h)} = \alpha^{anh},$$

$$\gamma^{J(h)} = \alpha^{b(2^m+1)h}.$$

Преобразуя выражение (4.40), получаем, что $(I(h), J(h))$-компонента $v(\phi)$ равна

$$T_2(\phi(\alpha^{anh})\alpha^{-b(2^m+1)h}).$$

Отсюда следует, что $f(X) = T_2(\phi(X^{an})X^{-b(2^m+1)})$ совпадает с $MS(v(\phi))$. Если $h(X) = \prod_{i \in \bar{R}_0} (X - \beta^i)$ — проверочный многочлен кода C_0, то, так как $\phi(X)$ имеет вид $\sum_{i \in R_0} S_i X^{2^m+1-i}$,

$S_{2i} = S_i^2$, получаем

$$f(X) = \sum_{j=0}^{m-1} \sum_{i \in \bar{R}_O} S_i^{2^j} X^{n(2^m+1)-(ani+b(2^m+1))2^j}.$$

Следовательно, корнями проверочного многочлена кода C являются элементы множества

$$\{\alpha^{(ani+b(2^m+1))2^j} \mid i \in \bar{R}_O,\ 0 \leqslant j < m\}.$$

Выбирая подходящим образом коды C_I, Берлекэмп и Юстесен показали существование большого числа кодов, для которых при достаточно больших их длинах отношение минимального расстояния к длине кода больше, чем у БЧХ-кодов, имеющих приблизительно такую же скорость передачи [51].

4.6.2. Коды Юстесена

Бесконечную последовательность систематических кодов $C_1, C_2, \ldots$ будем называть *асимптотически хорошей*, если выполнены следующие условия:

$$n_i < n_{i+1},$$
$$R_i \geqslant R,$$
$$d_i/n_i \geqslant \delta,$$

где n_i, R_i, d_i — соответственно длина, скорость и минимальное расстояние кода C_i, а $0 < R < 1$ и $0 < \delta < 1$ — некоторые числа. Из границы Варшамова — Гилберта следует существование асимптотически хорошей последовательности систематических кодов, параметры δ и R которой в двоичном случае удовлетворяют условию $H(\delta) < 1 - R$. Однако здесь утверждается только существование асимптотически хорошей последовательности кодов и ничего не говорится о том, как конкретно строить коды C_i. Если попытаться строить коды C_i, следуя выводу границы Варшамова — Гилберта, то нужно последовательно выбирать каждый столбец матрицы H. Юстесен предложил метод построения асимптотически хорошей последовательности кодов, который является столь же конструктивным, как и методы построения БЧХ-кодов или ОРМ-кодов [50]. Для простоты будем рассматривать двоичные коды. Код Юстесена является каскадным кодом, в качестве внешнего кода C_O которого используется $(2^m - 1, k_O)$-код Рида — Соломона над $GF(2^m)$. Положим $n_O = 2^m - 1$. Пусть α — примитивный элемент $GF(2^m)$, $\varphi(\beta)$ — представление элемента $\beta \in GF(2^m)$ в виде вектора над $GF(2)$ относительно некоторого фиксированного определенного базиса и $(\varphi(\beta_1), \varphi(\beta_2), \ldots, \varphi(\beta_l))$ — вектор длины lm, являющийся

последовательностью векторов $\varphi(\beta_1)$, $\varphi(\beta_2)$, ..., $\varphi(\beta_l)$. Множество двоичных векторов $\{(\varphi(v_0), \varphi(\alpha^0 v_0), \varphi(v_1), \varphi(\alpha^1 v_1), \ldots, \varphi(v_i), \varphi(\alpha^i v_i), \ldots, \varphi(v_{n_O-1}), \varphi(\alpha^{n_O-1} v_{n_O-1})) \mid (v_0, v_1, \ldots, v_{n_O-1}) \in C_O\}$ называется *кодом Юстесена*. Очевидно, что код Юстесена является линейным $(2m(2^m-1), k_O m)$-кодом. Как и раньше, кодовые векторы будем представлять в виде матриц размера $n_O \times 2m$, $(i+1)$-й строкой которых является $(\varphi(v_i), \varphi(\alpha^i v_i))$. Строка матрицы является нулевой только при $v=0$. Любые две ненулевые строки $(\varphi(v_i), \varphi(\alpha^i v_i))$ и $(\varphi(v_j), \varphi(\alpha^j v_j))$ обладают тем свойством, что если $v_i = v_j$, то

$$(\varphi(v_i), \varphi(\alpha^i v_i)) \neq (\varphi(v_j), \varphi(\alpha^j v_j)),$$

так как $\alpha^i v_i \neq \alpha^j v_j$ (здесь $v_i \neq 0$, $0 \leqslant i < j < n_O$). Минимальный вес кода C_O, согласно примеру 4.4, равен $n_O + 1 - k_O$. Следовательно, если только кодовый вектор является ненулевым, то по меньшей мере $n_O + 1 - k_O$ его строк также являются ненулевыми, а значит, и различными. Поэтому минимальный вес кода Юстесена не меньше суммы весов $n_O + 1 - k_O$ ненулевых двоичных векторов длины $2m$ минимально возможного веса, а именно

$$d_J \geqslant \sum_{j=1}^{w} j \binom{2m}{j} + (w+1)\left(n_O - k_O + 1 - \sum_{j=1}^{w} \binom{2m}{j}\right), \quad (4.41)$$

где $\sum_{j=1}^{w+1} \binom{2m}{j} \geqslant n_O - k_O + 1 \geqslant \sum_{i=1}^{w} \binom{2m}{j}$.

Пусть $m > 2$ и $\lambda = H^{-1}((1/2) - (1/\log 2m))$, где H^{-1} — функция, обратная энтропии[1]). Из леммы 1.1, положив в ней $q=2$, получаем

$$\sum_{j=1}^{2\lambda m} \binom{2m}{j} \leqslant 2^{2mH(\lambda)} = 2^{m - 2m/\log 2m}.$$

Тогда

$$\rho = \sum_{j=1}^{2\lambda m} \binom{2m}{j} \Big/ (n_O - k_O + 1) \leqslant \frac{1}{1-2R}\left(1 + \frac{1}{2^m - 1}\right) 2^{-2m/\log 2m},$$

где $R = k_O/[2(2^m-1)]$. Заметим, что параметр R является скоростью рассматриваемого кода. Из формул (4.41) и (4.42) следует, что

$$d_J \geqslant (n_O - k_O + 1)(1-\rho)2m\lambda > 2mn_O(1-2R)(1-\rho)\lambda >$$
$$> 2mn_O(1-2R)(1-\rho)H^{-1}((1/2) - (1/\log 2m)).$$

[1]) Для простоты $\log_2 2m$ обозначается здесь через $\log 2m$.

Поскольку длина кода n равна $2mn_0$ и ρ стремится к нулю с ростом m при фиксированной скорости R, то

$$\frac{d_J}{n} \geqslant H^{-1}\left(\frac{1}{2}\right)(1-2R)(1-O_1(m)) \approx 0{,}11\,(1-2R), \quad (4.43)$$

где $O_1(m) \to 0$ при $m \to \infty$ и фиксированном R. Это означает, что бесконечная последовательность кодов Юстесена (при $m \to \infty$) является асимптотически хорошей. Однако правая часть формулы (4.43) значительно меньше границы Варшамова — Гилберта $H^{-1}(1-R)$.

Описанные выше коды имели скорость $0 < R < 1/2$. Однако если модифицировать описанную выше конструкцию, а именно в качестве $(i+1)$-й строки кодовых матриц использовать векторы $(\varphi(v_i), \varphi^{(s)}(\alpha^i v_i))$, где $\varphi^{(s)}(\beta)$ — вектор длины s, компонентами которого являются первые s компонент $\varphi(s)$, то получается линейный $((m+s)(2^m-1), mk_0)$-код, который может иметь скорость $R > 1/2$. Используя ту же технику, что и раньше, можно найти следующую нижнюю оценку для минимального расстояния модифицированных кодов:

$$\frac{d_J}{n} \geqslant \left(1 - \frac{R}{\sigma}\right) H^{-1}(1-\sigma)(1-O_2(m)),$$

где $\sigma = m/(2m-s)$, а $O_2(m) \to 0$ при $m \to \infty$.

Методы декодирования кодов Юстесена описаны в работе [50]. Как указывалось в разд. 3.3, рассматривавшиеся выше асимптотические свойства кодов представляют главным образом теоретический интерес. Одной из нерешенных проблем является построение последовательности кодов, которые можно было бы задать столь же *конструктивно*, как и коды Юстесена, но которые лежали бы на границе Варшамова — Гилберта. Кроме того, открытым остается вопрос о существовании асимптотически хороших последовательностей циклических кодов.

4.7. Коды Гоппы

4.7.1. Определение

Коды Гоппы представляют собой очень широкий класс линейных кодов, включающий все БЧХ-коды. Известно, что среди этих кодов есть коды, лежащие на границе Варшамова — Гилберта. Однако такие коды не задаются конструктивно. Для кодов Гоппы в области «конструктивных расстояний» известны алгебраические методы декодирования, подобные методам декодирования БЧХ-кодов. Коды Гоппы — это один из важнейших классов линейных кодов, не являющихся, вообще говоря, циклическими.

При рассмотрении кодов Гоппы будем следовать работе [53], ограничившись случаем двоичных кодов [1]). Пусть $0 < n \leqslant 2^m$ и $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ — множество элементов поля $GF(2^m)$. Обозначим через V_n n-мерное векторное пространство над $GF(2)$. Вектору $v = (v_1, v_2, \ldots, v_n) \in V_n$ сопоставим рациональную функцию переменного z (z будет принимать значения в $GF(2^m)$):

$$R_v(z) = \sum_{i=1}^{n} \frac{v_i}{z - \alpha_i}.$$

Пусть $g(z)$ — многочлен от z с коэффициентами из $GF(2^m)$, среди корней которого нет элементов из L. *Код Гоппы*, задаваемый парой (L, g), определяется следующим образом:

$$\{v \mid R_v(z) \equiv 0 \pmod{g(z)},\ v \in V_n\}. \tag{4.44}$$

Многочлен $g(z)$ называется многочленом Гоппы. Очевидно, что введенный код является линейным. Пусть $\{i_1, i_2, \ldots, i_l\}$ — множество индексов i компонент v_i вектора $v = (v_1, v_2, \ldots, v_n) \in V_n$, равных единице. Пусть

$$f_v(z) = (z - \alpha_{i_1})(z - \alpha_{i_2}) \ldots (z - \alpha_{i_l}). \tag{4.45}$$

Тогда, как указывалось в разд. 2.6.6, $R_v(z)$ можно представить в виде

$$R_v(z) = f_v'(z)/f_v(z), \tag{4.46}$$

где f_v' — производная f_v.

Для любого $\alpha_i \in L$ справедливо следующее соотношение [2]):

$$\frac{1}{z - \alpha_i} \equiv \frac{g(z) - g(\alpha_i)}{z - \alpha_i} g^{-1}(\alpha_i) \pmod{g(z)}. \tag{4.47}$$

Заметим, что правая часть последнего уравнения является многочленом степени $\deg g - 1$ ($\deg g$ — степень многочлена g).

Используя (4.47), условие $R_v(z) \equiv 0 \pmod{g(z)}$ из (4.44) можно переписать в следующем виде:

$$\sum_{i=1}^{n} v_i \frac{g(z) - g(\alpha_i)}{z - \alpha_i} g^{-1}(\alpha_i) \equiv 0 \pmod{g(z)}.$$

Поскольку левая часть этого сравнения является многочленом степени $\deg g - 1$ или менее, то оно эквивалентно выполнению

---

[1]) Случай $q > 2$ рассмотрен в работе [55].

[2]) Так как используется поле $GF(2^m)$, то знак — ниже может заменяться на знак +. Поскольку $g(\alpha_i) \neq 0$, то $z - \alpha_i$ и $g(z)$ являются взаимно простыми многочленами, так что $1/(z - \alpha_i)$ — элемент, обратный $z - \alpha_i$ в кольце многочленов по модулю $g(z)$.

следующего равенства:

$$\sum_{i=1}^{n} v_i \frac{g(z)-g(\alpha_i)}{z-\alpha_i} g^{-1}(\alpha_i)=0. \tag{4.48}$$

Пусть $g(z)=\sum_{i=0}^{t} b_i z^i$ и $\deg g = t$. Без ограничения общности будем считать, что $b_t = 1$. Приравнивая суммы коэффициентов при $z^{t-1}, z^{t-2}, \ldots, z$, 1 в (4.48) нулю, получим систему из t линейных уравнений относительно $v_1, v_2, \ldots, v_n$. Матрица коэффициентов этой системы имеет вид

$$\begin{bmatrix} b_t g^{-1}(\alpha_1) & \ldots & b_t g^{-1}(\alpha_n) \\ (b_{t-1}+b_t\alpha_1)g^{-1}(\alpha_1) & \ldots & (b_{t-1}+b_t\alpha_n)g^{-1}(\alpha_n) \\ \vdots & & \vdots \\ (b_1+b_2\alpha_1+\ldots+b_t\alpha_1^{t-1})g^{-1}(\alpha_1) & \ldots & (b_1+b_2\alpha_n+\ldots+b_t\alpha_n^{t-1})g^{-1}(\alpha_n) \end{bmatrix}. \tag{4.49}$$

Из этой матрицы с помощью операций над строками (учитывая, что $b_t = 1$) можно получить следующую матрицу:

$$H=\begin{bmatrix} g^{-1}(\alpha_1) & \ldots & g^{1}(\alpha_n) \\ \alpha_1 g^{-1}(\alpha_1) & \ldots & \alpha_n g^{-1}(\alpha_n) \\ \vdots & & \vdots \\ \alpha_1^{t-1}g^{-1}(\alpha_1) & \ldots & \alpha_n^{t-1}g^{-1}(\alpha_n) \end{bmatrix}=$$

$$=\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^{t-1} & \alpha_2^{t-1} & \ldots & \alpha_n^{t-1} \end{bmatrix} \begin{bmatrix} g^{-1}(\alpha_1) & 0 & \ldots & 0 \\ 0 & g^{-1}(\alpha_2) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & g^{-1}(\alpha_n) \end{bmatrix}. \tag{4.50}$$

Как следует из вышеизложенного, H — проверочная матрица (заметим, однако, что если каждый элемент этой матрицы представить в виде вектор-столбца над $GF(2)$, то строки полученной таким образом матрицы над $GF(2)$, вообще говоря, могут оказаться линейно зависимыми). Следовательно, если k — число информационных символов этого кода, то

$$n-k \leqslant m \deg g. \tag{4.51}$$

В частности, если $n \mid 2^m - 1$, β — элемент порядка n поля $GF(2^m)$, $L = \{\beta^0, \beta^1, \ldots, \beta^{n-1}\}$ и $g(z) = z^t$, то

$$H = \begin{bmatrix} 1 & \beta^{-t} & \beta^{-2t} & \ldots & \beta^{-(n-1)t} \\ 1 & \beta^{-(t-1)} & \beta^{-2(t-1)} & \ldots & \beta^{-(n-1)(t-1)} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & \beta^{-1} & \beta^{-2} & \ldots & \beta^{-(n-1)} \end{bmatrix}.$$

Если заменить β^{-1} на β и поменять порядок следования строк на противоположный, то можно легко увидеть, что эта матрица является не чем иным, как проверочной матрицей БЧХ-кода с $l = 0$. Следовательно, класс кодов Гоппы включает БЧХ-коды с $l = 0$.

Пусть $\bar{g}(z)$ — многочлен минимальной степени, являющийся полным квадратом [т. е. имеющий вид $\varphi^2(z)$], такой, что $g(z) \mid \bar{g}(z)$. Если v — вектор, принадлежащий коду Гоппы, то, согласно формулам (4.44) и (4.46),

$$f_v'(z)/f_v(z) \equiv 0 \pmod{g(z)}.$$

Поэтому

$$f_v'(z) \equiv 0 \pmod{g(z)}. \tag{4.52}$$

Так как рассматривается поле характеристики 2, то многочлен f_v' состоит только из слагаемых четных степеней и, значит, является полным квадратом. Следовательно,

$$f_v'(z) \equiv 0 \mod \bar{g}(z). \tag{4.53}$$

Так как $w(v) = \deg f_v$, то из последнего соотношения получаем оценку снизу для минимального веса $d_{\text{мин}}$:

$$d_{\text{мин}} \geqslant \deg \bar{g} + 1. \tag{4.54}$$

Если все корни $g(z)$ различны, то $\bar{g} = g^2$ и в этом случае

$$d_{\text{мин}} > 2 \deg g + 1. \tag{4.55}$$

Для задания кода Гоппы нужно задать m, L и g. Однако код Гоппы можно также определить, если вместо g задать произвольным образом вектор $a = (a_1, a_2, \ldots, a_n)$, $a_i \in GF(2^m) - \{0\}$, $1 \leqslant i \leqslant n$. Действительно, в этом случае многочлен $g(z)$ минимальной степени, такой, что $g^{-1}(\alpha_i) = a_i$ $(1 \leqslant i \leqslant n)$ определяется единственным образом. Матрица H кода Гоппы, зада-

ваемого этим многочленом, имеет вид

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \alpha_1^{t-1} & \alpha_2^{t-1} & \dots & \alpha_n^{t-1} \end{bmatrix} \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & a_n \end{bmatrix}.$$

Утверждение 4.27 [53]. *Пусть* β — *элемент порядка* n *поля* $GF(2^m)$, *и пусть* $L = \{\beta^1, \beta^2, \dots, \beta^n\}$. Если *код Гоппы, определяемый этим множеством* L, *является циклическим, то он представляет собой БЧХ-код.*

Идея доказательства. Пусть v — произвольный кодовый вектор, и пусть $v^{(i)}$ — вектор, получающийся из v i-кратным циклическим сдвигом. Положим $f_v(z) = \prod_{j=1}^{l} (z - \beta^{j_l})$. По определению

$$f_v^{(i)}(z) = \prod_{j=1}^{l} (z - \beta^{t+i_l}) = \beta^{il} f_v(\beta^{-i} z).$$

Так как при всех $0 \leqslant i < n$ вектор $v^{(i)}$ является кодовым вектором, то из формулы (4.52) получаем

$$f_v'^{(i)}(z) \equiv 0 \pmod{g(z)}.$$

Следовательно,

$$f_v'(z) \equiv 0 \pmod{g(\beta^i z)}.$$

Если $g(z)$ имеет своим корнем ненулевой элемент γ, то n различных элементов $\gamma, \gamma\beta, \gamma\beta^2, \dots, \gamma\beta^{n-1}$ должны быть корнями f_v', а это противоречит тому, что $\deg f_v' < n$. Следовательно, $g(z) = z^t$. Но тогда, как уже указывалось выше, код Гоппы является БЧХ-кодом.

Пример 4.15 [56]. Пусть α — примитивный элемент $GF(2^m)$ и $L = \{\alpha^1, \alpha^2, \dots, \alpha^{n-1}, 0\}$. В этом случае длина кода n равна 2^m и многочлен Гоппы $g(z)$ не имеет корней в $GF(2^m)$. В частности, рассмотрим случай, когда $g(z)$ является неприводимым многочленом над $GF(2^m)$ степени t. Корни $g(z)$ в этом случае принадлежат $GF(2^{mt})$. Из формул (4.51) и (4.55) имеем

$$n - k \leqslant mt,$$

$$d_{\text{мин}} \geqslant 2t + 1.$$

Аналогичные оценки имеют место и для параметров примитивных БЧХ-кодов длины $2^m - 1$, однако длина кодов Гоппы на единицу больше.

Предположим, что для представления кодового вектора $v = (v_1, v_2, \ldots, v_n)$ используется многочлен $c_v(X)$ над $GF(2^m)$, такой, что

$$c_v(\alpha^i) = v_i, \qquad 1 \leqslant i < n,$$
$$c_v(0) = v_n.$$

Далее вместо векторного представления будем использовать представление многочленами (для простоты многочлен $c_v(X)$ будем обозначать через $c(X)$). Используя проверочную матрицу (4.50), получаем

$$\sum_{X \in GF(2^m)} \frac{X^i c(X)}{g(X)} = 0, \qquad 0 \leqslant i < t. \tag{4.56}$$

Рассмотрим расширение этого кода Гоппы. Обозначим добавленный символ индексом ∞. Полагая $\bar{L} = L \cup \{\infty\}$, имеем

$$\sum_{X \in \bar{L}} c(X) = 0. \tag{4.57}$$

Если условиться, что в формуле (4.56) $X^i/g(X) = 0$ при $X = \infty$, то можно считать, что X в (4.56) пробегает все элементы $\bar{L}$. Но тогда, согласно равенству (4.57),

$$\sum_{X \in \bar{L}} \frac{g(X)\, c(X)}{g(X)} = 0. \tag{4.58}$$

Из формул (4.56) и (4.58) имеем

$$\sum_{X \in \bar{L}} \frac{X^i c(X)}{g(X)} = 0, \qquad 0 \leqslant i \leqslant t. \tag{4.59}$$

Пусть $h(X)$ — многочлен, получаемый из $g(X)$ с помощью аффинного преобразования $X \to (AX + B)/(CX + D)$ $(A, B, C, D \in GF(2^m), AD \neq BC)$ множества $GF(2^m) \cup \{\infty\}$ в себя. Тогда, как следует из вышеизложенного, расширенный код Гоппы, задаваемый многочленом Гоппы $h(X)$ и тем же множеством L, может быть получен из расширенного кода Гоппы, задаваемого многочленом Гоппы $g(X)$, с помощью того же аффинного преобразования множества двоичных символов[1]). Если многочлен $g(X)$ инвариантен относительно аффинного преобразования T_A, то расширенный код Гоппы, задаваемый многочленом Гоппы $g(X)$, тоже является инвариантным относительно этого аффинного преобразования множества двоичных символов. Так как

[1]) Так как между множеством двоичных символов и множеством $GF(2^m) \cup \{\infty\}$ имеется взаимно однозначное соответствие, то преобразование множества $GF(2^m) \cup \{\infty\}$ можно рассматривать как преобразование соответствующего множества двоичных символов.

в случае $t = 2$, согласно утверждению 2.12, произвольный неприводимый многочлен может быть получен из любого другого неприводимого многочлена с помощью аффинного преобразования (с точностью до постоянного множителя), то все расширенные коды Гоппы с параметрами $n = 2^m + 1$ и $t = 2$ попарно эквивалентны. Без ограничения общности (см. утверждение 2.10) будем считать, что $g(z) = z^2 + z + \beta$ и $T_2(\beta) = 1$. Если γ — корень уравнения $g(z) = 0$, то $\gamma + 1$ и β/γ также являются корнями этого уравнения. Следовательно, код инвариантен относительно аффинных преобразований

$$\rho_1\colon X \to X + 1,$$

$$\rho_2\colon X \to \beta/X.$$

Покажем, что при соответствующем выборе элемента β преобразование $\rho = \rho_2\rho_1\colon X \to \beta/(X+1)$ является циклической перестановкой множества $\bar{L}$. Действительно, как известно, в общем случае для любого преобразования σ множества $\bar{L}$ существует такое разбиение $\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_l$ множества $\bar{L}$, что σ может быть представлено как произведение циклических преобразований $\sigma_1, \sigma_2, \ldots, \sigma_l$ множеств $\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_l$ соответственно. Если n' — число элементов, входящих в $\bar{L}_1$, то преобразование $\sigma^{n'}$ оставляет на месте элементы $\bar{L}_1$. Следовательно, для того чтобы показать, что σ является циклическим преобразованием, а именно что $l = 1$, следует показать, что ни одно из преобразований σ^i, $0 < i < 2^m + 1$, не имеет неподвижных точек. Необходимым условием того, что аффинное преобразование $(AX + B)/(CX + D)$ имеет неподвижную точку, является существование среди элементов множества $\bar{L}$ решения уравнения $(AX + B)/(CX + D) = X$. Как следует из утверждения 4.28, это условие эквивалентно тому, что собственные значения λ_1 и λ_2 матрицы $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ принадлежат $GF(2^m)$. С другой стороны, если положить

$$\sigma\colon (AX + B)/(CX + D),$$

$$\sigma'\colon (A'X + B')/(C'X + D'),$$

$$\sigma\sigma'\colon (A''X + B'')/(C''X + D''),$$

то

$$\begin{bmatrix} A'' & B'' \\ C'' & D'' \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} A' & B' \\ C' & D' \end{bmatrix}.$$

Следовательно, необходимым условием того, что σ^i имеет стационарную точку, является принадлежность собственных зна-

чений λ_1^i и λ_2^i матрицы $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^i$ полю $GF(2^m)$[1]). Итак, согласно вышеизложенному, чтобы показать, что ρ является циклическим преобразованием, следует доказать, что $\lambda^i \in GF(2^m) \Leftrightarrow i \equiv 0 \pmod{2^m+1}$, где λ — собственное значение матрицы $\begin{bmatrix} 0 & \beta \\ 1 & 1 \end{bmatrix}$ (λ является корнем уравнения $z^2+z+\beta=0$ и принадлежит $GF(2^{2m})$). Пусть γ — элемент $GF(2^{2m})$ порядка 2^m+1, и пусть $\beta_1=\gamma+\gamma^{-1}$. Так как $\gamma^{2^m}=\gamma^{-1}$, то $\beta_1^{2^m}=\beta_1$, т. е. $\beta_1 \in GF(2^m)$. Если записать равенство $\beta_1=\gamma+\gamma^{-1}$ в виде

$$(\beta_1^{-1}\gamma)^2+\beta_1^{-1}\gamma+\beta_1^{-2}=0$$

и положить $\beta=\beta_1^{-2}$, то уравнение $z^2+z+\beta=0$ будет иметь своими корнями элементы $\beta_1^{-1}\gamma$, $\beta_1^{-1}\gamma^{-1}$. Тогда $\beta_1^{-i}\gamma^{\pm i} \in GF(2^m) \Leftrightarrow \gamma^{\pm i} \in GF(2^m)$. Так как порядок 2^m+1 элемента γ и 2^m-1 являются взаимно простыми числами, то $\gamma^{\pm i} \in GF(2^m) \Leftrightarrow i \equiv 0 \pmod{2^m+1}$. Таким образом, при подходящем выборе β преобразование ρ оказывается циклическим и, следовательно, в этом случае расширенные коды Гоппы эквивалентны циклическим кодам.

Утверждение 4.28. *Необходимым условием существования среди элементов множества L решения уравнения $(AX+B)/(CX+D)=X$ является принадлежность корня уравнения*

$$\begin{vmatrix} A-z, & B \\ C, & D-z \end{vmatrix} = 0 \text{ полю } GF(2^m).$$

Краткое доказательство. Если $C=0$, то утверждение очевидно. В случае $C \neq 0$ без ограничения общности можно положить $C=1$. В этом случае первое уравнение принимает вид $X^2+(A+D)X+B=0$, а второе $z^2+(A+D)z+B+AD=0$. При $A+D=0$ оба эти уравнения имеют корни в $GF(2^m)$. Предположим, что $A+D \neq 0$. Если вместо X и z ввести соответственно $(A+D)^{-1}X$ и $(A+D)^{-1}z$, то придем к следующим уравнениям: $X^2+X+(A+D)^{-2}B=0$ и $z^2+z+(A+D)^{-2}\times$ $\times(AD+B)=0$. Так как $[A/(A+D)]^2+A/(A+D)=AD/(A+D)^2$, то $T_2((A+D)^{-2}AD)=0$. Поэтому

$$T_2\big((A+D)^{-2}AD+(A+D)^{-2}B\big)=T_2\big((A+D)^{-2}B\big).$$

---

[1]) Как хорошо известно, собственные значения матрицы $\begin{bmatrix} A & B \\ C & D \end{bmatrix}^i$ равны i-м степеням собственных значений матрицы $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ [5].

Но тогда, как следует из утверждений 2.10 и 2.11, если одно из уравнений имеет корень в $GF(2^m)$, то и другое также имеет корень в $GF(2^m)$.

Утверждение 4.29 [54]. *Рассмотрим подкласс кодов Гоппы длины* $n = 2^m$, *для которых* $L = \mathrm{GF}(2^m)$, а многочлены Гоппы $g(z)$ — *неприводимые многочлены над* $GF(2^m)$. *Для любого* $0 < R < 1/2$, *любого целого* k, $n/k \geqslant R$ *в этом подклассе существует код Гоппы с любым наперед заданным минимальным расстоянием* d, *удовлетворяющим неравенству*

$$H(d/n) \leqslant [(n-k)/n](1 - O(n)).$$

Идея доказательства. Доказательство почти аналогично доказательству утверждений 3.19 и 3.20.

4.7.2. Метод декодирования

Пусть v и e — соответственно переданный кодовый вектор и вектор ошибок. Принятый вектор $v + e$ обозначим через u. Положим $u = (u_1, u_2, \ldots, u_n)$. Определяя многочлен $f_u(z)$ с помощью формулы (4.45), из соотношения (4.46) получаем

$$\frac{f'_u(z)}{f_u(z)} = \frac{f'_v(z)}{f_v(z)} + \frac{f'_e(z)}{f_e(z)}.$$

Так как f_v не имеет кратных корней, то, согласно формуле (4.43),

$$\frac{f'_v(z)}{f_v(z)} \equiv 0 \pmod{\bar{g}(z)}.$$

Следовательно,

$$\frac{f'_e(z)}{f_e(z)} \equiv \frac{f'_u(z)}{f_u(z)} \pmod{\bar{g}(z)}.$$

Положим

$$s(z) = \sum_{i=1}^{n} u_i \frac{\bar{g}(z) - \bar{g}(\alpha_i)}{z - \alpha_i} \bar{g}^{-1}(\alpha_i). \tag{4.60}$$

Многочлен $s(z)$ может быть найден по принятому вектору. Имеет место следующее соотношение:

$$s(z) \equiv \frac{f'_u(z)}{f_u(z)} \equiv \frac{f'_e(z)}{f_e(z)} \pmod{\bar{g}(z)}.$$

Так как $\deg g$ — четное число, то положим $\deg \bar{g} = 2t$. Далее будем предполагать, что $0 < w(e) \leqslant t$, или, что то же самое,

$0 < \deg f_e \leqslant t$. Задача заключается в том, чтобы найти по заданному многочлену $s(z) \neq 0$ такой многочлен $f_e(z)$, что

$$f'_e(z) \equiv f_e(z)\, s(z) \pmod{\bar{g}(z)}.$$

Сначала покажем, что многочлен, являющийся решением этой задачи, определяется однозначно с точностью до свободного члена. Пусть $\phi(z)$ — многочлен с коэффициентами из $GF(2^m)$, не имеющий кратных корней. Если предположить, что $\deg \phi \leqslant t$ и

$$\phi'(z) \equiv \phi(z)\, s(z) \pmod{\bar{g}(z)},$$

то $\phi'(z) \equiv \phi(z)\,(f'_e(z)/f_e(z)) \pmod{\bar{g}(z)}$, а следовательно, $(\phi(z) f_e(z))' \equiv 0 \pmod{\bar{g}(z)}$. Так как $\deg(\phi(z) f_e(z))' \leqslant 2t$, то $(\phi(z) f_e(z))' = 0$, так что $\phi'(z) f_e(z) = \phi(z) f'_e(z)$. Так как $f_e(z)$ не имеет кратных корней, то $f_e(z)$ и $f'_e(z)$ взаимно простые, а поэтому $f_e(z) \mid \phi(z)$. Аналогично получаем, что $\phi(z) \mid f_e(z)$ и, значит, $f_e(z) = c\phi(z)$, $c \in GF(2^m)$.

Далее для простоты рассмотрим лишь случай, когда $g(z)$ не имеет кратных корней. При этом, определяя $s(z)$ с помощью равенства (4.60), предварительно заменив в нем $\bar{g}$ на g, получим

$$f'_e(z) \equiv f_e(z)\, s(z) \pmod{g(z)}.$$

Если $\phi(z)$ — многочлен степени не больше t, такой, что

$$\phi'(z) \equiv \phi(z)\, s(z) \pmod{g(z)}, \tag{4.61}$$

то $(\phi f_e)' \equiv 0 \pmod{g(z)}$. Так как $(\phi f_e)'$ является полным квадратом, то $(\phi f_e)' \equiv 0 \pmod{g^2(z)}$. Следовательно, $(\phi f_e)' = 0$ и с точностью до свободного члена решение определяется однозначно. Решение сравнения (4.61) имеет следующий вид:

$$\phi(z) = a_0 + z\varphi(z), \quad \deg \varphi < t, \quad a_0 \in GF(2).$$

Подставляя это решение в формулу (4.61), после несложных преобразований получаем

$$(z\varphi'(z) + \varphi(z))(1 + zs(z)) \equiv a_0 s(z) \pmod{g(z)}. \tag{4.62}$$

Пусть $\varphi(z) = \sum_{i=0}^{t-1} \varphi_i z^i$; $s(z) = \sum_{j=0}^{t-1} s_j z^j$; $(1 + zs(z)) z^i \equiv \sum_{j=0}^{t-1} c_{ji} z^j \pmod{g(z)}$, $0 \leqslant i < t$; D — матрица размера $t \times t$, у которой диагональные элементы с четными номерами равны 1, а остальные элементы равны 0; T_s — матрица размера $t \times t$, (i, j)-й элемент которой равен $c_{j-1,\, i-1}$; $\bar{\varphi} = (\varphi_0, \varphi_1, \ldots, \varphi_{t-1})$; $\bar{s} = (s_0, s_1, \ldots, s_{t-1})$. В этих обозначениях уравнение (4.62) будет иметь следующий вид:

$$(D + T_s)\, \bar{\varphi}^T = a_0 \bar{s}^T. \tag{4.63}$$

Из единственности решения следует, что матрица $D+T_s$ является невырожденной, если $a_0=1$, и вырожденной, если $a_0=0$. Таким образом, мы получили следующий алгоритм декодирования. (Если $0\in L$, будем считать, что $\alpha_n=0$.)

1) По принятому вектору находится $s(z)$. Если $s(z)=0$, то принимается решение о том, что ошибок при передаче не было. Если $s(z)\neq 0$, то переходят к шагу 2.

2) Для каждого $0\leqslant i<t$ находится остаток $\sum_{j=0}^{t-1} c_{ij}z^j$ от деления $(1+zs(z))z^i$ на $g(z)$.

3) Строится матрица $D+T_s$. Если полученная матрица является вырожденной, то к n-й компоненте прибавляется единица и осуществляется переход к шагу 1. Если полученная матрица оказывается невырожденной, то находится решение $\bar{\varphi}$ уравнения

$$(D+T_s)\bar{\varphi}^T=\bar{s}^T$$

и далее определяется $\varphi(z)$.

4) Подставляя в $\phi(z)=1+z\varphi(z)$ последовательно элементы $L-\{0\}$, находят все корни $\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_\nu}\in L-\{0\}$ уравнения $\phi(z)=0$. Компоненты принятого вектора с номерами $i_1, i_2, \ldots, i_\nu$ заменяются на противоположные.

Как следует из утверждения 4.29, минимальный вес $d_{\text{мин}}$ «хороших» кодов Гоппы значительно превышает правые части (4.54) и (4.55). Однако хороших методов исправления ошибок веса более чем $t+1$ неизвестно. В тех случаях, когда ошибки веса до t исправляются, а ошибки большего веса обнаруживаются, вероятность ошибочного декодирования обычно тем меньше, чем больше $d_{\text{мин}}$, так что с этой точки зрения «хорошие» коды Гоппы более предпочтительны, чем примитивные БЧХ-коды, для которых $d_{\text{мин}}/d_{\text{БЧХ}}$ меньше приблизительно в полтора раза.

Пример 4.16 [53]. Пусть $m=4$ и $L=GF(2^4)$. Так как в этом случае многочлен $g(z)$ не должен иметь корней среди элементов $GF(2^4)$, то он должен выбираться из многочленов, неприводимых над $GF(2^4)$ или их произведений. Пусть α — примитивный элемент $GF(2^4)$, а X^4+X+1 — его минимальный многочлен. Упорядочим элементы L следующим образом: $L=\{\alpha^1, \alpha^2, \ldots, \alpha^{15}, 0\}$.

Так как элемент α^3 имеет порядок 5 и его минимальный многочлен есть $X^4+X^3+X^2+X+1$, то сумма корней $T_2(\alpha^3)$ равна 1. Тогда, согласно утверждению 2.10, многочлен $z^2+z+\alpha^3$ является неприводимым над $GF(2^4)$ и его можно использовать в качестве $g(z)$. С помощью таблицы из задачи 2.59 можно проверить, что $g(\alpha)=\alpha^2+\alpha+\alpha^3=\alpha^{11}$, $g^{-1}(\alpha)=\alpha^4$,

$g(\alpha^2)=\alpha^4+\alpha^2+\alpha^3=1+\alpha+\alpha^2+\alpha^3=\alpha^{12}$, $g^{-1}(\alpha^2)=\alpha^3$, ..., ..., $(1+\alpha)g^{-1}(\alpha)=\alpha^8$, $(1+\alpha^2)g^{-1}(\alpha^2)=\alpha^{11}$, ...
В этом случае матрица (4.49) имеет вид

$$\begin{bmatrix} \alpha^4 & \alpha^3 & \alpha^9 & \alpha^4 & \alpha & \alpha^8 & \alpha^6 & \alpha^3 & \alpha^6 & \alpha & \alpha^2 & \alpha^2 & \alpha^8 & \alpha^9 & \alpha^{12} & \alpha^{12} \\ \alpha^8 & \alpha^{11} & \alpha^8 & \alpha^5 & \alpha^{11} & \alpha^6 & 1 & \alpha^5 & \alpha^{13} & \alpha^6 & \alpha^{14} & \alpha^{13} & \alpha^{14} & \alpha^{12} & 0 & \alpha^{12} \end{bmatrix}.$$

Если представить эту матрицу в двоичном виде, то она будет иметь следующий вид:

$$\begin{bmatrix} 0&1&1&0&0&0&1&1&1&0&0&0&0&1&1&1 \\ 0&0&0&0&0&1&1&0&1&0&1&1&1&0&1&1 \\ 1&0&1&1&1&0&0&0&0&1&0&0&0&1&1&1 \\ 1&0&0&1&0&1&0&0&0&0&0&0&1&0&1&1 \\ 0&1&0&0&1&1&0&0&1&1&1&1&1&1&0&1 \\ 1&1&1&1&1&1&0&1&1&1&0&1&0&1&0&1 \\ 0&1&0&1&1&0&0&1&0&0&0&0&0&1&0&1 \\ 1&0&1&0&0&0&1&0&1&0&1&1&1&1&0&1 \end{bmatrix}.$$

Так как ранг этой матрицы равен 8, то код имеет 8 проверочных и 8 информационных символов.

Предположим, что ошибки возникли в 1-й и 5-й компонентах. Тогда

$$s(z)=(\alpha+\alpha^4)z+\alpha^8+\alpha^{11}=\alpha^7+z,$$

$$1+zs(z)=z^2+\alpha^7z+1\equiv(\alpha^7+1)z+\alpha^3+1 \pmod{g(z)}\equiv$$
$$\equiv\alpha^{14}+\alpha^9z \pmod{g(z)},$$

$$(1+zs(z))z\equiv\alpha^9z^2+\alpha^{14}z\equiv\alpha^9(z+\alpha^3)+\alpha^{14}z\equiv$$
$$\equiv(\alpha^9+\alpha^{14})z+\alpha^{12}\equiv\alpha^{12}+\alpha^4z \pmod{g(z)},$$

$$D+T_s=\begin{bmatrix}\alpha^{14} & \alpha^{12}\\ \alpha^9 & \alpha^4+1\end{bmatrix}=\begin{bmatrix}\alpha^{14} & \alpha^{12}\\ \alpha^9 & \alpha\end{bmatrix},$$

$\bar{s}=(\alpha^7, 1)$, $\bar{\varphi}=(\varphi_0, \varphi_1)$. Уравнения (4.59) имеют вид

$$\alpha^{14}\varphi_0+\alpha^{12}\varphi_1=\alpha^7,$$
$$\alpha^9\varphi_0+\alpha\varphi_1=1.$$

$$\varphi_0=(\alpha^8+\alpha^{12})/(1+\alpha^6)=\alpha^{11},\quad \varphi_1=(\alpha^{14}+\alpha)/(1+\alpha^6)=\alpha^9.$$

Следовательно, $\phi(z)=1+\alpha^{11}z+\alpha^9z^2$. В этом случае, как легко проверить, $\phi(\alpha)=\phi(\alpha^5)=0$. Из примера 4.15 следует, что расширенный код кода Гоппы, рассмотренного в данном примере, эквивалентен циклическому $(17, 8)$-коду с минимальным весом 6 (принадлежащему классу БЧХ-кодов из утверждения 4.4).

4.8. Коды, исправляющие пачки ошибок

Для любого линейного (n, k)-кода, исправляющего пачки ошибок длины b и меньше, должно выполняться следующее соотношение Рейджера [57] (утверждение 3.7):

$$n - k \geqslant 2b.$$

Важнейшими известными кодами, исправляющими пачки ошибок, являются коды, получающиеся из циклических кодов, укороченных циклических кодов или кодов Рида — Соломона. Рассмотрим сначала укороченные циклические коды (включая циклические коды). Пусть $g(X)$ — порождающий многочлен. Пачку ошибок длины b или менее можно представить в следующем виде:

$$X^i e(X),$$

где $e(0) \neq 0$, $\deg e < b$, $0 \leqslant i < n - \deg e$. Следовательно, для того чтобы код исправлял пачки ошибок длины b и менее, необходимо, чтобы для произвольных многочленов $e_1(X)$ и $e_2(X)$, таких, что $e_j(0) \neq 0$, $\deg e_j < b$, и любых целых чисел i_1 и i_2, $0 \leqslant i_j < n - \deg e_j$ $(1 \leqslant j \leqslant 2)$, выполнялось соотношение

$$X^{i_1} e_1(X) + X^{i_2} e_2(X) \not\equiv 0 \pmod{g(X)}.$$

Если последнее соотношение выполняется при $i_1 = 0$, то, как следует из свойств укороченных циклических кодов, оно выполняется и для произвольного i_1. Поэтому указанное выше условие эквивалентно тому, что

$$e_1(X) + X^i e_2(X) \equiv 0 \pmod{g(X)},$$
$$0 \leqslant i < n - \deg e_2.$$

В частности, в случае циклических кодов, если $n - \deg e_2 \leqslant i < n$, то $X^{n-i}(e_1(X) + X^i e_2(X)) \equiv e_2(X) + X^{n-i} e_1(X) \times (\mathrm{mod}\, X^n - 1)$, $n - i \leqslant \deg e_2 < b < n - b$, так что можно ограничиться следующими значениями i: $0 \leqslant i < n$. Следующая теорема дает так называемый *метод перемежения*, позволяющий по кодам небольшой длины строить более длинные коды [4].

Теорема 4.10. *Если (укороченный) циклический (n, k)-код C с порождающим многочленом $g(X)$ исправляет пачки ошибок длины b и менее, то (укороченный) циклический (nl, kl)-код $C^{(l)}$ с порождающим многочленом $g(X^l)$ исправляет пачки ошибок длины bl и менее.*

Доказательство. Так как C — циклический код, то $g(X) \mid X^n - 1$. Следовательно, $g(X^l) \mid X^{nl} - 1$, так что $C^{(l)}$ также является

циклическим кодом. Для любого многочлена $f(X)=\sum_{j=0}^{n-1} f_j X^j$ и любого h, $0\leqslant h<l$, имеет место равенство $f(X)=\sum_{h=0}^{l-1} X^h f^{(h)}(X^l)$, где $f^{(h)}(X)=\sum_{j=0}^{(n-h)/l} f_{h+lj}X^j$. Если многочлены $e_1(X)$, $e_2(X)$ и целые числа i, i_0 такие, что $e_1(0)\neq 0$, $\deg e_1<bl$, $e_2(0)\neq 0$, $\deg e_2<bl$, $0\leqslant il+i_0<nl-\deg e_2$, $0\leqslant i<n/l$, $0\leqslant i_0<l$, то

$$e_1(X)+X^{il+i_0}e_2(X)=\sum_{h=0}^{l-1} X^h\left[e_1^{(h)}(X^l)+X^{il}e_2^{(h-i_0)}(X^l)\right]$$

(здесь индекс $h-i_0$ у $e_2^{(h-i_0)}$ вычисляется по модулю l). Обозначим левую часть последнего равенства через $e(X)$. Заметим, что $g(X^l)\,|\,e(X)$ только в том случае, если $g(X^l)\,|\,(e_1^{(h)}(X^l)+X^{il}e_2^{(h-i_0)}(X^l))$, $0\leqslant h<l$. Следовательно, $g(X)\,|\,[e_1^{(h)}(X)+X^i e_2^{(h-i_0)}(X)]$. Поскольку это противоречит тому, что код C исправляет пачки ошибок длины b и менее, то

$$e(X)\not\equiv 0 \pmod{g(X^l)},$$

т. е. код $C^{(l)}$ исправляет пачки ошибок длины bl и менее.

Следующая теорема принадлежит Файру [9, 58].

Теорема 4.11. *Пусть C_0 — циклический код длины n_0 с порождающим многочленом $g_0(X)$, исправляющий пачки ошибок длины b и менее, и пусть $g_1(X)$ — неприводимый взаимно простой с $g_0(X)$ многочлен с периодом n_1, степень которого не меньше b. Тогда циклический код длины $n=n_0n_1/(n_0,n_1)$ с порождающим многочленом $g(X)=g_0(X)g_1(X)$ исправляет пачки ошибок длины b и менее.*

Доказательство. Предположим, что для многочленов $e_1(X)$, $e_2(X)$ и целого i, таких, что $e_1(0)\neq 0$,

$$\deg e_1<b,\quad e_2(0)\neq 0,\quad \deg e_2<b,\quad 0\leqslant i<n,$$

имеет место следующее сравнение:

$$e_1(X)+X^i e_2(X)\equiv 0 \pmod{g_0(X)g_1(X)}.$$

Пусть i_0 — остаток, а l — частное от деления i на n_0. Так как $g_0(X)\,|\,X^{n_0}-1$, то

$$e_1(X)+X^i e_2(X)\equiv e_1(X)+X^{i_0}e_2(X)\equiv$$
$$\equiv 0 \pmod{g_0(X)},\qquad 0\leqslant i_0<n_0.$$

Поскольку C_0 исправляет пачки ошибок длины b и менее, то $i_0=0$ и

$$e_1(X)=-e_2(X),$$

а поэтому

$$(X^{ln_0} - 1)\, e_1(X) \equiv 0 \pmod{g_0(X)\, g_1(X)}.$$

Так как $g_1(X)$ неприводим и $\deg g_1 \geqslant b > \deg e_1$, то

$$g_1(X) \mid X^{ln_0} - 1.$$

По предположению $g_0(X) \mid X^{n_0} - 1$ и $(g_0(X), g_1(X)) = 1$, так что

$$g(X) \mid X^{ln_0} - 1.$$

Поскольку $0 \leqslant i = \ln_0 < n$, то это противоречит тому, что n является минимальным целым числом, таким, что $g(X) \mid X^n - 1$.

Следствие 4.11. *Пусть $g_1(X)$ — неприводимый многочлен с периодом n_1, степень которого не меньше b, взаимно простой с $X^{2b} - 1$. Тогда циклический код длины $(2b-1)n_1/(2b-1, n_1)$ с порождающим многочленом $(X^{2b-1} - 1)\, g_1(X)$ исправляет пачки ошибок длины b и менее.*

Коды, введенные в этом следствии, называют *кодами Файра* [58]. Код Файра имеет более чем $3b - 1$ проверочных символов, что на $b - 1$ больше, чем нижняя граница Рейджера, равная $2b$. Для длин кодов до нескольких тысяч построены «вручную» или с помощью вычислительных машин таблицы порождающих многочленов лучших циклических и укороченных циклических кодов, исправляющих пачки ошибок [60—65].

Утверждение 4.30 [66]. *Пусть C — укороченный циклический код с порождающим многочленом $(X^l - 1)\, g_0(X)$, исправляющий пачки ошибок длины b и меньше. Если минимальный вес C равен $2t + 2$, то код C исправляет все случайные ошибки веса t и менее, а кроме того, все пачки ошибок длины $\min(b, l)$ и менее.*

Краткое доказательство. Если $t \geqslant \min(b, l)$, то утверждение очевидно. Предположим, что $t < \min(b, l)$. Пусть $e(X)$ — многочлен ошибок. Положим

$$e(X) \equiv r_1(X) \pmod{X^l - 1},$$
$$\deg r_1 < l.$$

1) Если вес $w(e(X))$ многочлена $e(X)$ не больше чем t, то

$$w(r_1(X)) \leqslant t.$$

2) Если $w(e(X)) > t$ или если $e(X)$ — пачка ошибок длины $\min(b, l)$ или менее, то

$$w(r_1(X)) > t.$$

Следовательно, остатки от деления векторов ошибок типа 1 и типа 2 на $(X^l - 1)\, g_0(X)$ различны, а поэтому принадлежат различным смежным классам.

Кодами, подобными кодам из утверждения 4.30, которые могут исправлять как случайные ошибки, так и пачки ошибок, являются коды, описанные ниже, а также коды, которые получаются, когда в качестве многочленов $g_0(X)$ в утверждении 4.30 используются порождающие многочлены БЧХ-кодов, а также некоторые другие многочлены [66, 67].

Вообще говоря, если в (n, k)-коде над $GF(q^m)$ элементы $GF(q^m)$ заменить на их представления в виде векторов размерности m над $GF(q)$, то можно получить (mn, mk)-код над $GF(q)$. Если исходный код линейный, то полученный таким образом код также будет линейным. Однако если исходный код является циклическим, то получающийся код может не быть циклическим. Возьмем в качестве исходного кода q^m-ичный (q^m-1, q^m-1-2t)-код Рида — Соломона, исправляющий t ошибок. Пусть C — q-ичный линейный $(m(q^m-1), m(q^m-1-2t))$-код, получающийся указанным выше способом. Непосредственно из построения кода C можно видеть, что он исправляет все пачки ошибок длины $m(t-1)+1$ и менее, а также все случайные ошибки веса t и менее. Аналогичные коды можно построить, если вместо кода Рида — Соломона использовать коды из примера 4.5.

Утверждение 4.31. *Описанный выше код C исправляет любые l или менее пачек ошибок длины $m((t/l)-1)+1$ и менее.*

5. Сверточные коды

I. Методы порогового декодирования

5.1. Общий обзор сверточных кодов

Сверточные (или рекуррентные) *коды* были введены Элайсом [1] из Массачусетского технологического института (МТИ). В отличие от блоковых кодов, когда каждый символ на выходе кодера зависит лишь от конечного числа информационных символов на входе кодера, в случае сверточных кодов символы на выходе кодера, вообще говоря, могут зависеть от бесконечного числа информационных символов. В этом смысле сверточные коды можно рассматривать как обобщение блоковых кодов.

Развитие теории сверточных кодов происходило в трех направлениях в соответствии с тремя важнейшими методами декодирования сверточных кодов: *метода порогового декодирования* [2], *метода декодирования по максимуму правдоподобия* [25] и *метода последовательного декодирования* [14]. Метод порогового декодирования сверточных кодов в принципе аналогичен методу мажоритарного декодирования блоковых кодов. Достоинством этого метода является простота алгоритма, а следовательно, и реализующих его устройств. Точно так же, как и для блоковых алгебраических кодов, число операций, необходимых для декодирования одного информационного символа, для этого алгоритма не превосходит некоторой постоянной величины. Метод декодирования по максимуму правдоподобия теоретически более эффективен, чем метод порогового декодирования, однако сложность устройств, необходимых для его реализации, возрастает экспоненциально с ростом длины кода. И наконец, метод последовательного декодирования является методом вероятностного декодирования, при котором число операций, необходимых для декодирования одного символа, является случайной величиной. При практически приемлемой сложности устройств метод последовательного декодирования по своим характеристикам приближается к методу декодирования по максимуму правдоподобия.

5.1.1. Метод порогового декодирования

Метод порогового декодирования, предложенный Месси [2], состоит в том, что при приеме вычисляются синдромы и эти синдромы или последовательности, полученные в результате линейного преобразования синдромов, подаются на входы поро-

гового элемента; исправление ошибок осуществляется далее с помощью символов, поступающих с выхода порогового элемента. Коды, которые допускают декодирование этим методом, должны обладать введенным Месси свойством ортогональности. Ряд кодов со скоростью $^1/_2$ и меньше, обладающих этим свойством, был построен Месси. В дальнейшем Робинсон и Бернстейн [3] на основе совершенных разностных множеств построили самоортогональные коды, для которых проверочные последовательности обладают свойством ортогональности и могут быть непосредственно поданы на входы пороговых элементов. К числу кодов, которые допускают пороговое декодирование, относятся также равномерные коды [36], аналоги кодов Рида — Маллера [2] и др.

Сверточные коды, исправляющие пачки ошибок, впервые были предложены Хегельбергером [4]. В дальнейшем Вайнер и Эш [6], Препарата [7], Берлекэмп [5], Месси [8] и др., основываясь главным образом на теории матриц, построили ряд квазиоптимальных кодов, исправляющих пачки ошибок и требующих для этого защитный интервал, близкий к минимально возможному. Ивадари [10—12], разлагая синдромы на простые конфигурации, построил класс кодов, допускающих очень простое кодирование и декодирование.

Для исправления случайных ошибок и пачек ошибок Коленбург разработал так называемые диффузные коды, допускающие пороговое декодирование [9]. Относительно регулярный метод построения самоортогональных диффузных кодов предложил Тонг [13].

Методы порогового декодирования основаны на простых идеях, и они находят широкое применение на практике. Работы в этой области оказали большое влияние на развитие исследований не только сверточных кодов, но и блоковых кодов, допускающих мажоритарное декодирование.

5.1.2. Методы последовательного декодирования

Последовательное декодирование было введено Возенкрафтом [14] раньше, чем пороговое декодирование, и на начальном этапе исследования в этой области велись главным образом в МТИ. Этот метод, приближающийся по своим характеристикам к методу декодирования по максимуму правдоподобия, требует для декодирования одного символа тем меньше операций, чем меньше уровень шума в канале; в отличие от этого при декодировании по максимуму правдоподобия число операций постоянно и достаточно велико, поскольку оно определяется максимальным уровнем шума в канале. Это свойство последовательного декодирования позволяет упростить требующиеся для его

реализации устройства. Предложенный Возенкрафтом алгоритм был положен в основу экспериментов по передаче данных по телефонным каналам, проводившихся в Линкольновской лаборатории МТИ и получивших название SECO [15].

Алгоритм Возенкрафта, предназначавшийся первоначально для двоичного симметричного канала, был обобщен Рейффеном [16] на случай произвольного дискретного *канала без памяти.* Крупным успехом исследований в этом направлении было открытие алгоритма Фано [17]. Среднее число операций, требующееся для декодирования одного символа при использовании алгоритма Фано, не зависит от кодового ограничения. Допуская простую техническую реализацию, а также обладая рядом других достоинств, этот алгоритм до настоящего времени остается одним из основных алгоритмов. Поскольку алгоритмы последовательного декодирования являются вероятностными, то анализ их работы связан с рядом сложностей. Один из путей преодоления этих сложностей заключается в том, чтобы, как и при доказательстве теоремы кодирования Шеннона для канала с шумом, найти характеристики алгоритма, средние по некоторому ансамблю кодов, и далее, воспользовавшись тем, что по крайней мере для одного из кодов в этом ансамбле характеристики алгоритма не хуже средних, сделать вывод об эффективности того или иного алгоритма. Метод определения границы случайного кодирования, предложенный Галлагером и подробно рассмотренный в гл. 1, также широко применяется для анализа алгоритмов последовательного декодирования. Этим методом вначале Севадж [18] нашел среднее число операций для алгоритма Фано; затем вновь Севадж [19], Джекобс и Берлекэмп [20] и др. показали, что распределение числа операций при декодировании является распределением Парето. Еще одним способом анализа сложных алгоритмов является их моделирование на вычислительной машине. Большая работа по моделированию алгоритмов последовательного декодирования была выполнена Джорданом [21] и др. Кроме этих методов для исследования поведения алгоритмов последовательного декодирования используются также качественные и общие методы.

При последовательном декодировании желательно использовать коды с большим минимальным расстоянием и по возможности меньшим кодовым ограничением. Удобно представлять коды в виде кодового дерева. Буссганг [22] опубликовал список кодов со скоростями $R = 1/2$ и $1/3$, которые могут использоваться в сочетании с последовательным декодированием, а Лин [23] предложил регулярный метод построения таких кодов. Кроме того, Костелло [24] предложил простой алгоритм, при использовании которого с помощью вычислительной машины можно строить коды с большим минимальным расстоянием.

При последовательном декодировании число операций, необходимых для декодирования одного символа, возрастает по мере увеличения уровня шума в канале. При сильном шуме принятые, но еще не декодированные символы накапливаются в буферной памяти декодера. В тех случаях, когда ожидающая обработки последовательность символов переполняет буферную память, декодер перестает работать правильно. Это явление характерно для последовательного декодирования и известно под названием переполнения буфера. Вероятность переполнения буфера можно вычислить или оценить с помощью распределения Парето для числа операций при декодировании, полученного Севаджем [18], а также Джекобсом и Берлекэмпом [20]. В действительности вероятность переполнения буфера часто оказывается гораздо больше, чем вероятность ошибочного декодирования, и именно она определяет возможности последовательного декодирования. Фальконер [38] предложил гибридный метод декодирования каскадных кодов, сочетающий в себе элементы алгебраического и последовательного декодирования. В качестве внешних кодов он использовал коды Рида — Соломона, а в качестве внутренних кодов — коды, допускающие последовательное декодирование. При этом в случае, если при последовательном декодировании число операций превышает некоторый определенный порог, то соответствующий символ внешнего кода считается стертым. Как показал Фальконер, такой метод декодирования позволяет уменьшить вероятность переполнения буфера.

Как уже указывалось выше, алгоритм Фано занимает центральное положение среди алгоритмов последовательного декодирования. Зигангировым [27] и Джелинеком [26] независимо был предложен другой алгоритм последовательного декодирования, известный под названием *стек-алгоритма*. Этот алгоритм во многих случаях лучше алгоритма Фано; он очень просто описывается и помогает проникнуть в существо различных принципиальных вопросов последовательного декодирования [1]).

5.1.3. Методы декодирования по максимуму правдоподобия

После введения Элайсом сверточных кодов встал вопрос о том, какие коды (сверточные или блоковые) потенциально лучше. Ответ на этот вопрос был дан Витерби [25], который получил верхние и нижние границы вероятности ошибки для свер-

[1]) Последовательному декодированию посвящена выпущенная в 1974 г. издательством «Связь» книга К. Ш. Зигангирова «Последовательное декодирование». В этой книге дан исторический обзор достижений по последовательному декодированию, объективно отражающий как советские, так и зарубежные работы. — *Прим. ред.*

точных кодов и с их помощью показал, что характеристики оптимальных сверточных кодов как функции кодового ограничения лучше соответствующих характеристик блоковых кодов той же длины. Для получения этих границ Витерби использовал алгоритм декодирования сверточных кодов, который, как оказалось, действует как алгоритм декодирования по максимуму правдоподобия. Этот алгоритм впоследствии стал называться *алгоритмом Витерби.*

Если на начальном этапе сверточные коды были лишь объектом теоретических исследований, то в дальнейшем они стали применяться сначала в сочетании с последовательным декодированием в космической связи, а затем в сочетании с пороговым декодированием в спутниковых системах передачи данных, системах морской радиосвязи и других системах. Кроме того, анализ структуры сверточных кодов связан с теорией автоматов. Несомненно, что в будущем будет наблюдаться дальнейшее углубление взаимосвязи между теоретическими исследованиями в области сверточных кодов и практическим использованием последних. В этой главе будут рассмотрены лишь методы порогового декодирования сверточных кодов и используемые при этом коды. Методам последовательного декодирования сверточных кодов посвящена следующая глава.

5.2. Представление сверточных кодов

Известно несколько способов представления сверточных кодов: представление с помощью оператора задержки, использованное Месси [2], представление с помощью матриц, использованное Вайнером и Эшем [6], представление с помощью дерева, обычно используемое при последовательном декодировании и др. Ответить на вопрос о том, какое из этих представлений является самым удобным, нельзя, поскольку все зависит от решаемой задачи. В данном разделе будут рассмотрены представление сверточных кодов с помощью оператора задержки и представление с помощью матриц, а также будет установлена связь между ними. Представление с помощью дерева будет описано в главе, посвященной последовательному декодированию.

5.2.1. Представление с помощью операторов задержки

На фиг. 5.1 показан сверточный кодер в общем виде. В каждый момент времени на вход кодера поступают k_0 информационных символов и с выходов декодера n_0 выходных символов посылаются в канал. Вообще говоря, символами могут быть элементы произвольного конечного поля $GF(q)$, и тогда информация передается со скоростью $R = (k_0/n_0)\log_2 q$ двоичных еди-

ниц на один выходной символ. Далее, если это не оговорено особо, предполагается, что все операции с входными и выходными символами кодера выполняются в поле $GF(q)$.

Удобный способ представления временных последовательностей предложен Хаффменом [31]; он основан на использовании оператора задержки D. При этом k_0 входных последовательностей информационных символов представляются в виде многочленов

$$I^{(j)}(D) = i_0^{(j)} + i_1^{(j)}D + i_2^{(j)}D^2 + \ldots, \quad j = 1, 2, \ldots, k_0, \tag{5.1}$$

где $i_u^{(j)}$ — информационный символ, поступивший на j-й вход кодера в момент u. Аналогично n_0 выходных последовательно-

Фиг. 5.1. Сверточный кодер.

стей представляются в виде многочленов

$$T^{(i)}(D) = t_0^{(i)} + t_1^{(i)}D + t_2^{(i)}D^2 + \ldots, \quad i = 1, 2, \ldots, n_0, \tag{5.2}$$

где $t_u^{(i)}$ — символ, появляющийся на i-м выходе кодера в момент u.

Связь между выходными последовательностями $T^{(i)}(D)$ и входными последовательностями $I^{(j)}(D)$ задается для некоторого m с помощью многочленов

$$G_{(i)}^{(j)}(D) = \sum_{k=0}^{m} g_{(i),\,k}^{(j)} D^k \tag{5.3}$$

степени m или меньше с коэффициентами из поля $GF(q)$. Эта связь выражается равенством

$$T^{(i)}(D) = \sum_{j=1}^{k} G_{(i)}^{(j)}(D)\, I^{(j)}(D). \tag{5.4}$$

Многочлены $G_{(i)}^{(j)}(D)$ называются порождающими многочленами; выбор порождающих многочленов $G_{(i)}^{(j)}(D)$ полностью определяет сверточный код.

Пример 5.1. Рассмотрим показанный на фиг. 5.2 кодер сверточного кода со скоростью $1/2$, имеющий один вход и два выхода. Если входная последовательность

$$I^{(1)}(D) = i_0^{(1)} + i_1^{(1)}D + i_2^{(1)}D^2 + \ldots$$

задана, то две выходные последовательности

$$T^{(1)}(D) = t_0^{(1)} + t_1^{(1)}D + t_2^{(1)}D^2 + \ldots,$$
$$T^{(2)}(D) = t_0^{(2)} + t_1^{(2)}D + t_2^{(2)}D^2 + \ldots$$

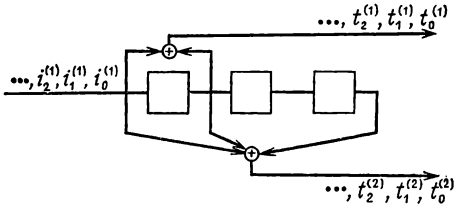
определяются порождающими многочленами

$$G_{(1)}^{(1)}(D) = 1 + D,$$
$$G_{(2)}^{(1)}(D) = 1 + D + D^3$$

следующим образом:

$$T^{(1)}(D) = G_{(1)}^{(1)}(D)\, I^{(1)}(D),$$
$$T^{(2)}(D) = G_{(2)}^{(1)}(D)\, I^{(1)}(D).$$

Если m — наибольшая из степеней порождающих многочленов, то произвольный фиксированный информационный символ



Фиг. 5.2. Кодер сверточного кода со скоростью $R = 1/2$.

может оказывать влияние на выходную последовательность в течение не более чем $m + 1$ единиц времени. За это время кодер выдаст всего $(m + 1)n_0$ символов, а поэтому говорят, что кодовое ограничение n_A этого кода составляет $(m + 1)n_0$ символов.

По аналогии с систематическими блоковыми кодами можно рассматривать также систематические сверточные коды. В случае систематических кодов первые k_0 выходных последовательностей совпадают с входными последовательностями:

$$T^{(i)}(D) = I^{(j)}(D), \qquad i = j = 1, 2, \ldots, k_0. \tag{5.5}$$

Остальные $n_0 - k_0$ выходных последовательностей, которые можно рассматривать как проверочные последовательности, являются линейными комбинациями входных последовательностей, т. е.

$$T^{(i)}(D) = \sum_{j=0}^{k_0} G_{(i)}^{(j)}(D)\, I^{(j)}(D) = G_{(i)}^{(1)}(D)\, I^{(1)}(D) + G_{(i)}^{(2)}(D)\, I^{(2)}(D) + \ldots$$
$$\ldots + G_{(i)}^{(k_0)}(D)\, I^{(k_0)}(D), \qquad i = k_0 + 1, \ldots, n_0. \tag{5.6}$$

Другими словами, каждый передаваемый по каналу символ проверочной последовательности является линейной комбинацией предшествующих информационных символов. Систематический сверточный код определяется уже выбором $k_0(n_0 - k_0)$ порождающих многочленов $G_{(i)}^{(j)}(D)$, $i = k_0 + 1, \ldots, n_0$, $j = 1, \ldots, k_0$.

Проверочные последовательности сверточного кода при приеме могут быть найдены следующим образом. Принятые последовательности $R^{(i)}(D)$, $i = 1, 2, \ldots, n_0$, могут быть представлены с помощью оператора задержки в виде

$$R^{(i)}(D) = r_0^{(i)} + r_1^{(i)} D + r_2^{(i)} D^2 + \ldots, \qquad i = 1, 2, \ldots, n_0, \tag{5.7}$$

где $r_u^{(i)}$ — символ i-й принятой последовательности, поступивший в момент времени u. Точно так же могут быть представлены и шумовые последовательности $E^{(i)}(D)$:

$$E^{(i)}(D) = e_0^{(i)} + e_1^{(i)} D + e_2^{(i)} D^2 + \ldots, \qquad i = 1, 2, \ldots, n_0, \tag{5.8}$$

где $e_u^{(i)}$ — шумовой символ, оказавшийся в момент u в i-й принятой последовательности. При этом

$$R^{(i)}(D) = T^{(i)}(D) + E^{(i)}(D), \qquad i = 1, 2, \ldots, n_0. \tag{5.9}$$

Произвольные суммы шумовых символов, вычисляемые при приеме, для сверточных кодов так же, как и в случае блоковых линейных кодов, будем называть проверками или проверочными суммами. По аналогии с блоковыми линейными кодами последовательности коэффициентов многочленов

$$S^{(i)}(D) = [G_{(i)}^{(1)}(D) R^{(1)}(D) + \ldots + G_{(i)}^{(k_0)}(D) R^{(k_0)}(D)] - R^{(i)}(D), \tag{5.10}$$
$$i = k_0 + 1, \ldots, n_0,$$

будем называть *синдромами*. Заметим, что синдромы являются последовательностями проверочных сумм, которые могут быть вычислены при приеме. Синдромами будем также иногда называть и сами многочлены (5.10). Подставляя выражения (5.6) и (5.9) в (5.10), получаем

$$S^{(i)}(D) = [G_{(i)}^{(1)}(D) E^{(1)}(D) + \ldots + G_{(i)}^{(k_0)}(D) E^{(k_0)}(D)] - E^{(i)}(D), \tag{5.11}$$
$$i = k_0 + 1, \ldots, n_0.$$

Если синдромы $S^{(i)}(D)$ представить в виде

$$S^{(i)}(D) = s_0^{(i)} + s_1^{(i)} D + s_2^{(i)} D^2 + \ldots, \qquad i = k_0 + 1, \ldots, n_0, \tag{5.12}$$

где $s_u^{(i)}$ — проверочная сумма последовательности $s^{(i)}(D)$, вычисляемая в момент u, то с учетом формулы (5.10) имеем

$$s_u^{(i)} = [g_{(i),\,u}^{(1)} e_0^{(1)} + g_{(i),\,u-1}^{(1)} e_1^{(1)} + \ldots + g_{(i),\,0}^{(1)} e_u^{(1)}] + \ldots$$
$$\ldots + [g_{(i),\,u}^{(k_0)} e_0^{(k_0)} + g_{(i),\,u-1}^{(k_0)} e_1^{(k_0)} + \ldots + g_{(i),\,0}^{(k_0)} e_u^{(k_0)}] - e_u^{(i)}. \tag{5.13}$$

Поскольку при рассмотренном выше методе кодирования передаваемые последовательности и синдромы получаются в результате свертки порождающих многочленов соответственно с информационными последовательностями и шумовыми последовательностями, то рассматриваемые коды получили название сверточных.

5.2.2. Представление с помощью матриц

Рассмотрим, каким образом можно задавать сверточные коды с помощью матриц. Символ $t_u^{(i)}$ i-й передаваемой последовательности, поступающий на вход канала в момент u, согласно формуле (5.4), определяется равенством

$$\begin{aligned} t_u^{(i)} &= [g_{(i),\,u}^{(1)} i_0^{(1)} + g_{(i),\,u-1}^{(1)} i_1^{(1)} + \ldots + g_{(i),\,0}^{(1)} i_u^{(1)}] + \ldots \\ &\quad \ldots + [g_{(i),\,u}^{(k_0)} i_0^{(k_0)} + g_{(i),\,u-1}^{(k_0)} i_1^{(k_0)} + \ldots + g_{(i),\,0}^{(k_0)} i_u^{(k_0)}] = \\ &= [g_{(i),\,u}^{(1)} i_0^{(1)} + \ldots + g_{(i),\,u}^{(k_0)} i_0^{(k_0)}] + \\ &\quad + [g_{(i),\,u-1}^{(1)} i_1^{(1)} + \ldots + g_{(i),\,u-1}^{(k_0)} i_1^{(k_0)}] + \ldots \\ &\quad \ldots + [g_{(i),\,0}^{(1)} i_u^{(1)} + \ldots + g_{(i),\,0}^{(k_0)} i_u^{(k_0)}]. \end{aligned} \tag{5.14}$$

Следовательно, вектор-строка из символов передаваемых последовательностей

$$\begin{aligned} \boldsymbol{T} &= (t_0^{(1)}, t_0^{(2)}, \ldots, t_0^{(n_0)}, t_1^{(1)}, t_1^{(2)}, \ldots, t_1^{(n_0)}, \ldots, t_m^{(1)}, t_m^{(2)}, \ldots, t_m^{(n_0)}, \ldots) = \\ &= (\boldsymbol{t}_0, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_m, \ldots), \end{aligned} \tag{5.15}$$

где

$$\boldsymbol{t}_u = (t_u^{(1)}, t_u^{(2)}, \ldots, t_u^{(n_0)}), \qquad u = 0, 1, \ldots, m, \ldots, \tag{5.15'}$$

и вектор-строка из символов информационных последовательностей

$$\begin{aligned} \boldsymbol{I} &= (i_0^{(1)}, i_0^{(2)}, \ldots, i_0^{(k_0)}, i_1^{(1)}, i_1^{(2)}, \ldots, i_1^{(k_0)}, \ldots, i_m^{(1)}, i_m^{(2)}, \ldots, i_m^{(k_0)}, \ldots) = \\ &= (\boldsymbol{i}_0, \boldsymbol{i}_1, \ldots, \boldsymbol{i}_m, \ldots), \end{aligned} \tag{5.16}$$

где

$$\boldsymbol{i}_u = (i_u^{(1)}, i_u^{(2)}, \ldots, i_u^{(k_0)}), \qquad u = 0, 1, \ldots, m, \ldots, \tag{5.16'}$$

связаны между собой следующим матричным равенством

$$\boldsymbol{T} = \boldsymbol{I} G_\infty, \tag{5.17}$$

где G_∞ — полубесконечная матрица, которая задается с помощью матриц

$$G_u = \begin{bmatrix} g^{(1)}_{(1),\,u} & \cdots & g^{(1)}_{(n_0),\,u} \\ \vdots & & \vdots \\ g^{(k_0)}_{(1),\,u} & \cdots & g^{(k_0)}_{(n_0),\,u} \end{bmatrix} \tag{5.18}$$

и имеет следующий вид:

$$G_\infty = \begin{bmatrix} G_0, & G_1, \ldots, G_m & . & \\ . & G_0, G_1, \ldots, G_m & . & \mathbf{0} \\ \mathbf{0} & . & G_0, G_1, \ldots, G_m & . \end{bmatrix}. \tag{5.19}$$

(Обозначенные символом **0** области матрицы G_∞ состоят полностью из нулей.) Матрица G_∞ называется порождающей матрицей сверточного кода.

В случае систематических кодов k_0 информационных символов входят в первые k_0 передаваемых последовательностей без изменений, так что порождающая матрица G_∞ систематического сверточного кода задается матрицами

$$G'_0 = \begin{bmatrix} 1 & 0 \ldots 0 & g^{(1)}_{(k_0+1),\,0} & \cdots & g^{(1)}_{(n_0),\,0} \\ 0 & 1 \ldots 0 & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & \ldots 1 & g^{(k_0)}_{(k_0+1),\,0} & \cdots & g^{(k_0)}_{(n_0),\,0} \end{bmatrix} = [\boldsymbol{I},\ G''_0],$$

$$G'_u = \begin{bmatrix} 0 & 0 \ldots 0 & g^{(1)}_{(k_0+1),\,u} & \cdots & g^{(1)}_{(n_0),\,u} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 \ldots 0 & g^{(k_0)}_{(k_0+1),\,u} & \cdots & g^{(k_0)}_{(n_0),\,u} \end{bmatrix} = [0,\ G''_u], \qquad u \geqslant 1$$

(здесь $\boldsymbol{I}$ и $\mathbf{0}$ — соответственно единичная и нулевая матрицы размера $k_0 \times k_0$) и имеет следующий вид:

$$G_\infty = \begin{bmatrix} . & G'_0 & G'_1 \ldots & G'_m & . & \\ & . & G'_0 \ldots & G'_m & . & \mathbf{0} \\ \mathbf{0} & & . & & . & \end{bmatrix}. \tag{5.19'}$$

Пример 5.2. Порождающая матрица кода, рассмотренного в примере 5.1, имеет вид

$$G_\infty = \begin{bmatrix} 11 & 11 & 00 & 01 & 00 & \dots & \dots \\ 00 & 11 & 11 & 00 & 01 & 00 & \dots & \dots \\ 00 & 00 & 11 & 11 & 00 & 01 & 00 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}.$$

В случае систематических кодов символы синдромов задаются равенством (5.13), которое можно записать в виде

$$s_u^{(i)} = [g_{(i),\,u}^{(1)} e_0^{(1)} + g_{(i),\,u}^{(2)} e_0^{(2)} + \dots + g_{(i),\,u}^{(k_0)} e_0^{(k_0)}] + [g_{(i),\,u-1}^{(1)} e_1^{(1)} + \dots + g_{(i),\,u-1}^{(k_0)} e_1^{(k_0)}] + \dots \dots + [g_{(i),\,0}^{(1)} e_u^{(1)} + \dots + g_{(i),\,0}^{(k_0)} e_u^{(k_0)}] - e_u^{(i)}. \quad (5.20)$$

Если ввести вектор-столбцы $\boldsymbol{S}_\infty$ и $\boldsymbol{E}_\infty$:

$$\boldsymbol{S}_\infty^T = [s_0^{(k_0+1)}, s_0^{(k_0+2)}, \dots, s_0^{(n_0)}, \dots, s_m^{(k_0+1)}, s_m^{(k_0+2)}, \dots, s_m^{(n_0)}, \dots],$$

$$\boldsymbol{E}_\infty^T = [e_0^{(1)}, e_0^{(2)}, \dots, e_0^{(n_0)}, e_1^{(1)}, e_1^{(2)}, \dots, e_1^{(n_0)}, \dots \dots, e_m^{(1)}, e_m^{(2)}, \dots, e_m^{(n_0)}, \dots],$$

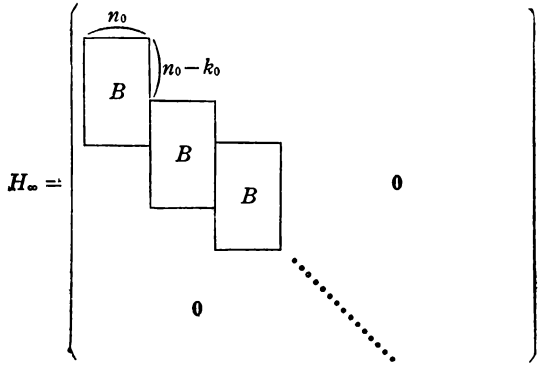
где T — символ транспонирования, то равенство (5.20) можно записать в матричных обозначениях следующим образом:

$$\boldsymbol{S}_\infty = H_\infty \boldsymbol{E}_\infty, \quad (5.21)$$

где H_∞ — полубесконечная матрица, которая задается матрицей

$$B = \begin{bmatrix} g_{(k_0+1),\,0}^{(1)} & g_{(k_0+1),\,0}^{(2)} & \dots & g_{(k_0+1),\,0}^{(k_0)} & -1 & 0 & \dots & 0 \\ g_{(k_0+2),\,0}^{(1)} & g_{(k_0+2),\,0}^{(2)} & \dots & g_{(k_0+2),\,0}^{(k_0)} & 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_{(n_0),\,0}^{(1)} & g_{(n_0),\,0}^{(2)} & \dots & g_{(n_0),\,0}^{(k_0)} & 0 & 0 & \dots & -1 \\ g_{(k_0+1),\,1}^{(1)} & g_{(k_0+1),\,1}^{(2)} & \dots & g_{(k_0+1),\,1}^{(k_0)} & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_{(n_0),\,1}^{(1)} & g_{(n_0),\,1}^{(2)} & \dots & g_{(n_0),\,1}^{(k_0)} & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_{(k_0+1),\,m}^{(1)} & g_{(k_0+1),\,m}^{(2)} & \dots & g_{(k_0+1),\,m}^{(k_0)} & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_{(n_0),\,m}^{(1)} & g_{(n_0),\,m}^{(2)} & \dots & g_{(n_0),\,m}^{(k_0)} & 0 & 0 & \dots & 0 \end{bmatrix} \quad (5.22)$$

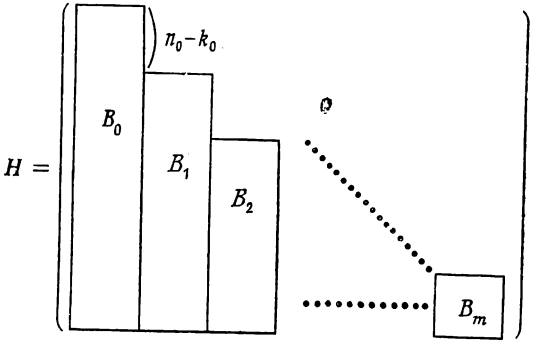
и имеет следующий вид:



$$H_\infty = \qquad (5.23)$$

Матрица H_∞ называется проверочной матрицей систематического сверточного кода и, как и в случае блоковых кодов, полностью задает код.

Ошибка, возникшая в некотором символе в момент времени 0, может оказывать влияние на символы синдромов лишь до момента m. Поэтому при изучении влияния ошибок, возникших в момент времени 0, можно не рассматривать символы синдромов в момент времени $m+1$ и далее, а ограничиться начальными участками синдромов и шумовых последовательностей, а именно рассматривать лишь совокупности из первых $(m+1)$ символов синдромов и шумовых последовательностей. Такие укороченные синдромы, состоящие из первых $(m+1)$ символов, называются *усеченными синдромами*. Проверочная матрица H, соответствующая таким усеченным синдромам и шумовым последовательностям, имеет вид



$$H = \qquad (5.24)$$

где B_i — матрица, получающаяся из матрицы B удалением нижних $i(n_0-k_0)$ строк $(i=0, 1, \ldots, m)$. Заметим, что для усе-

ченной передаваемой последовательности $\boldsymbol{T}_t$ имеет место равенство

$$H\boldsymbol{T}_t = 0, \tag{5.25}$$

так что при приеме, умножая усеченную принимаемую последовательность

$$\boldsymbol{R}_t^T = [t_0^{(1)} + e_0^{(1)},\ t_0^{(2)} + e_0^{(2)},\ \ldots,\ t_0^{(n_0)} + e_0^{(n_0)},\ t_1^{(1)} + e_1^{(1)},\ \ldots$$
$$\ldots,\ t_1^{(n_0)} + e_1^{(n_0)},\ \ldots,\ t_m^{(1)} + e_m^{(1)},\ t_m^{(2)} + e_m^{(2)},\ \ldots,\ t_m^{(n_0)} + e_m^{(n_0)}] \tag{5.26}$$

на матрицу H, можно вычислить укороченный синдром $\boldsymbol{S}_t$, поскольку

$$H\boldsymbol{R}_t = H(\boldsymbol{T}_t + \boldsymbol{E}_t) = H\boldsymbol{T}_t + H\boldsymbol{E}_t = \boldsymbol{S}_t. \tag{5.27}$$

В некоторых случаях удобнее пользоваться векторами

$$\boldsymbol{S}_\infty^T = [s_0^{(k_0+1)},\ \ldots,\ s_m^{(k_0+1)},\ \ldots,\ s_0^{(k_0+2)},\ \ldots,\ s_m^{(k_0+2)},\ \ldots$$
$$\ldots,\ s_0^{(n_0)},\ \ldots,\ s_m^{(n_0)},\ \ldots], \tag{5.28}$$

$$\boldsymbol{E}_\infty^T = [e_0^{(1)},\ \ldots,\ e_m^{(1)},\ \ldots,\ e_0^{(2)},\ \ldots,\ e_m^{(2)},\ \ldots,\ e_0^{(n_0)},\ \ldots,\ e_m^{(n_0)},\ \ldots]. \tag{5.29}$$

При этом

$$\boldsymbol{S}_\infty = [H_\infty^\Delta :\ -I_\infty]\boldsymbol{E}_\infty, \tag{5.30}$$

где I_∞ — полубесконечная единичная матрица,

$$H_\infty^\Delta = \begin{bmatrix} H^\Delta_{1,\,k_0+1} & H^\Delta_{2,\,k_0+1} & \ldots & H^\Delta_{k_0,\,k_0+1} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ H^\Delta_{1,\,n_1} & H^\Delta_{2,\,n_0} & \ldots & H^\Delta_{k_0,\,n_0} \end{bmatrix}, \tag{5.31}$$

$$H_{ji}^\Delta = \begin{bmatrix} g^{(j)}_{(i),\,0} & & & \\ \cdot & & & \\ & \cdot & & \\ & & \cdot & \\ g^{(j)}_{(i),\,1} & g^{(j)}_{(i),\,0} & & \\ \cdot & \cdot & & \\ \cdot & & \cdot & \\ & & & \cdot \\ g^{(j)}_{(i),\,m} & & g^{(j)}_{(i),\,0} & \\ \cdot & & & \cdot \\ \cdot & & & \cdot \end{bmatrix}.$$

Матрицы H_{ji}^Δ обычно называют проверочными треугольниками [1]).

[1]) Проверочными треугольниками авторы иногда называют также матрицы $[H_\infty^\Delta :\ I_\infty]$. — *Прим. перев.*

Пример 5.3. Двоичный систематический сверточный код со скоростью $R = k_0/n_0 = {}^2/_3$ и кодовым ограничением $n_A = (m+1)n_0 = 12$, задаваемый порождающими многочленами

$$G^{(1)}_{(3)}(D) = 1 + D),$$

$$G^{(2)}_{(3)}(D) = 1 + D^2 + D^3,$$

имеет порождающую матрицу

$$G_\infty = \begin{bmatrix} 1\,0\,1 & 0\,0\,1 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0 & \cdots\cdots \\ 0\,1\,1 & 0\,0\,0 & 0\,0\,1 & 0\,0\,1 & 0\,0\,0 & \cdots\cdots \\ & 1\,0\,1 & 0\,0\,1 & 0\,0\,0 & 0\,0\,0 & 0\,0\,0\ \cdots \\ & 0\,1\,1 & 0\,0\,0 & 0\,0\,1 & 0\,0\,1 & 0\,0\,0\ \cdots \\ & \ddots & \ddots & \ddots & \ddots & \end{bmatrix}.$$

В этом случае

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

и

$$[H^{\Delta}_\infty : I_\infty] = \begin{bmatrix} 1 & & & 1 & & & 1 & & & \\ 1\,1 & & & 0\,1 & & & & 1 & & \\ 0\,1\,1 & & \mathbf{0} & 1\,0\,1 & & \mathbf{0} & & & 1 & & \mathbf{0} \\ 0\,0\,1\,1 & & & 1\,1\,0\,1 & & & & & & 1 & \\ 0\,0\,0\,1\,1 & & & 0\,1\,1\,0 & & & & & & & 1 \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \end{bmatrix}.$$

Выше рассматривались синдромы с момента времени 0 до момента времени m. В то же время иногда возникает необходимость рассматривать синдромы с момента времени $-t$ до момента времени m. В этих случаях можно достаточно просто обобщить порождающие многочлены, порождающие матрицы и проверочные матрицы, заменив индекс 0, показывающий время, на $-t$.

Пример 5.4. Если код, задаваемый в интервале времени от 0 до 3 проверочным треугольником

$$H^{\Delta}=\begin{bmatrix} 1 & & & & \\ 1 & 1 & & & 0 \\ 1 & 1 & 1 & & \\ 0 & 1 & 1 & 1 & \end{bmatrix},$$

рассматривать в интервале времени от —3 до 3, то проверочная матрица будет иметь следующий вид:

$$H^{\Delta}=\begin{bmatrix} 1 & 1 & 1 & & & & \\ & 1 & 1 & 1 & & 0 & \\ & & 1 & 1 & 1 & & \\ & 0 & & 1 & 1 & 1 & \\ & & & & 1 & 1 & 1 \end{bmatrix}.$$

Число ненулевых элементов проверочных и порождающих матриц сверточных кодов по сравнению с числом нулевых элементов обычно сравнительно невелико. Поэтому такие матрицы можно просто задавать, указывая для каждого столбца лишь номера строк, в которых расположены ненулевые элементы этого столбца.

Пример 5.5. Следуя указанному выше способу, матрицу B из примера 5.3 можно просто задать следующим образом:

$$B=\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}=\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & \\ & 4 & \end{bmatrix}. \qquad (5.32)$$

Это представление особенно удобно для матриц больших размеров.

5.3. Пример порогового декодирования

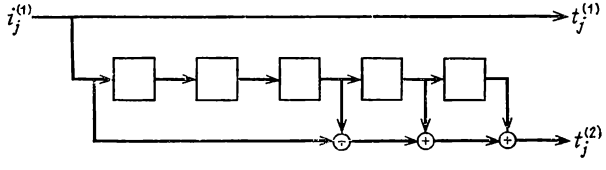
Пороговое декодирование сверточных кодов основано на тех же принципах, что и мажоритарное декодирование блоковых кодов. Прежде чем рассмотреть эти принципы в общем случае, приведем простой пример, попутно вводя необходимые понятия.

Пример 5.6. Рассмотрим сверточный код со скоростью $R = 1/2$ и кодовым ограничением $n_A = 12$, исправляющий любые

две случайные ошибки в пределах кодового ограничения и имеющий следующий проверочный треугольник:

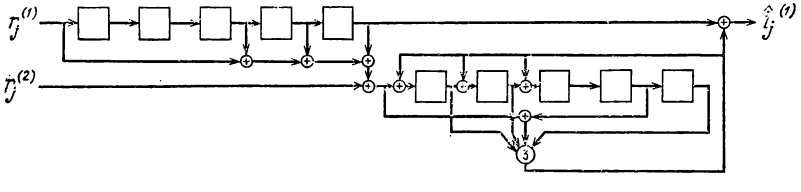
$$[H^{\Delta}, I_6] = \begin{bmatrix} 1 & & & & & & 1 & & & & & \\ 0 & 1 & & & & & & 1 & & & & \\ 0 & 0 & 1 & & \mathbf{0} & & & & 1 & & \mathbf{0} & \\ 1 & 0 & 0 & 1 & & & & & & 1 & & \\ 1 & 1 & 0 & 0 & 1 & & & & & & 1 & \\ 1 & 1 & 1 & 0 & 0 & 1 & & & & & & 1 \end{bmatrix}. \qquad (5.33)$$

Кодер и декодер этого кода показаны соответственно на фиг. 5.3 и фиг. 5.4. Каждый информационный двоичный символ посту-



Фиг. 5.3. Кодер ($R = 1/2$, $J = 4$).

пает в кодер с левой стороны, выходит из кодера с правой стороны и передается по каналу. В это же время в кодере формируется проверочный символ, который также передается по каналу. Этот кодер, согласно вводимой в гл. 7 терминологии, называется Rn_A-разрядным кодером. Декодер по принятой информационной последовательности сначала вычисляет точно так же, как и при кодировании, проверочные символы. Для вычисления



Фиг. 5.4. Декодер ($R = 1/2$, $J = 4$).

этих проверочных символов декодер содержит Rn_A-разрядный кодер. Далее, полученная с помощью этого кодера проверочная последовательность складывается покомпонентно по модулю 2 с принятой проверочной последовательностью, и в результате получается синдром

$$S^T = [s_0^{(2)}, s_1^{(2)}, s_2^{(2)}, \ldots, s_5^{(2)}].$$

Символы этого синдрома подаются в регистр сдвига, показанный на фиг. 5.4 в нижней части декодера. Согласно формуле (5.33), синдром $\boldsymbol{S}$ и шумовые символы $e_u^{(i)}$, $u = 0, \ldots, 5$, $i = 1, 2$, связаны между собой следующим матричным соотношением:

$$\boldsymbol{S} = \begin{bmatrix} s_0^{(2)} \\ s_1^{(2)} \\ s_2^{(2)} \\ s_3^{(2)} \\ s_4^{(2)} \\ s_5^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & & & & & & 1 & & & & & \\ 0 & 1 & & & & & 1 & & 0 & & & \\ 0 & 0 & 1 & & 0 & & & 1 & & & & \\ 1 & 0 & 0 & 1 & & & & & 1 & & & \\ 1 & 1 & 0 & 0 & 1 & & & & & 1 & & \\ 1 & 1 & 1 & 0 & 0 & 1 & & & & & 1 & \end{bmatrix} \begin{bmatrix} e_0^{(1)} \\ e_1^{(1)} \\ e_2^{(1)} \\ e_3^{(1)} \\ e_4^{(1)} \\ e_5^{(1)} \\ e_0^{(2)} \\ e_1^{(2)} \\ e_2^{(2)} \\ e_3^{(2)} \\ e_4^{(2)} \\ e_5^{(2)} \end{bmatrix}, \tag{5.34}$$

или, что то же самое,

$$\begin{aligned} s_0^{(2)} &= e_0^{(1)} + e_0^{(2)}, \\ s_1^{(2)} &= e_1^{(1)} + e_1^{(2)}, \\ s_2^{(2)} &= e_2^{(1)} + e_2^{(2)}, \\ s_3^{(2)} &= e_0^{(1)} + e_3^{(1)} + e_3^{(2)}, \\ s_4^{(2)} &= e_0^{(1)} + e_1^{(1)} + e_4^{(1)} + e_4^{(2)}, \\ s_5^{(2)} &= e_0^{(1)} + e_1^{(1)} + e_2^{(1)} + e_5^{(1)} + e_5^{(2)}. \end{aligned} \tag{5.35}$$

С помощью линейного преобразования синдрома $\boldsymbol{S}$ находятся величины A_v, $v = 1, 2, 3, 4$:

$$\begin{aligned} A_1 &= s_0^{(2)} = e_0^{(1)} + e_0^{(2)}, \\ A_2 &= s_3^{(2)} = e_0^{(1)} + e_3^{(1)} + e_3^{(2)}, \\ A_3 &= s_4^{(2)} = e_0^{(1)} + e_1^{(1)} + e_4^{(1)} + e_4^{(2)}, \\ A_4 &= s_5^{(2)} + s_1^{(2)} = e_0^{(1)} + e_2^{(1)} + e_5^{(1)} + e_1^{(2)} + e_5^{(2)}. \end{aligned} \tag{5.36}$$

Линейное преобразование, с помощью которого по синдрому $\boldsymbol{S}$ находятся величины A_v, называется *правилом ортогонализации*,

а получаемые с его помощью величины A_v называются *составными проверками*. Если посмотреть на составные проверки (5.36), то можно заметить, что они обладают следующим свойством: шумовой символ $e_0^{(1)}$, воздействующий на информационный символ в момент 0, входит во все составные проверки A_v, $v = 1, 2, 3, 4$, и в то же время любой другой шумовой символ входит не более чем в одну из этих проверок. Такая совокупность составных проверок $\{A_v\}$ называется *ортогональной* относительно $e_0^{(1)}$.

Если $e_0^{(1)} = 1$ и остальные шумовые символы равны нулю, то составные проверки, задаваемые формулами (5.36), равны 1, а именно $A_1 = A_2 = A_3 = A_4 = 1$. Если же один из шумовых символов, отличных от $e_0^{(1)}$, равен 1, то среди проверок по крайней мере три будут 1. И наконец, если $e_0^{(1)} = 0$ и среди остальных шумовых символов имеется не более чем две единицы, то среди проверок A_v равными единице будут самое большее две проверки. Если эти составные проверки подать на вход порогового элемента с порогом 2,5, то символ на выходе последнего будет равен $e_0^{(1)}$ всегда, когда число случайных ошибок в пределах кодового ограничения не больше двух. Далее, складывая символ с выхода порогового элемента с принятым двоичным символом $i_0^{(1)} + e_0^{(1)}$, который хранится в ячейке памяти, показанной на фиг. 5.4 в верхней части декодера, шумовой символ $e_0^{(1)}$ всегда можно исправить. При этом выходной символ порогового элемента по цепи обратной связи подается на схему вычисления синдрома и корректирует последний, устраняя влияние $e_0^{(1)}$. Такая процедура называется *декодированием с обратной связью*. Следовательно, аналогичные рассуждения можно провести и для моментов времени $u = 1, 2, \ldots$.

Если шумы не выходят за пределы допустимых, то исправление ошибок осуществляется правильно. Если шумы превышают допустимую величину, то могут возникнуть естественные ошибки декодирования. Однако поскольку декодер содержит цепь обратной связи, то однократная ошибка декодирования приводит к искажению синдрома и дальнейшим ошибкам декодирования. Это явление характерно для сверточных кодов и известно как *распространение ошибок*. Конечно, всегда хотелось бы использовать коды, не способствующие распространению ошибок.

5.4. Принцип порогового декодирования

Равенство (5.13), которое задает символы синдромов в моменты времени $u \geqslant 0$, запишем в несколько более общем виде, считая, что символы синдромов порождаются в моменты вре-

мени u от $-t$ до m:

$$s_u^{(i)} = [g_{(i),\,u+t}^{(1)} e_{-t}^{(1)} + g_{(i),\,u+t-1}^{(1)} e_{-t+1}^{(1)} + \ldots + g_{(i),\,u-m}^{(1)} e_m^{(1)}] + \ldots$$
$$\ldots + [g_{(i),\,u+t}^{(k_0)} e_{-t}^{(k_0)} + g_{(i),\,u+t-1}^{(k_0)} e_{-t+1}^{(k_0)} + \ldots$$
$$\ldots + g_{(i),\,u-m}^{(k_0)} e_m^{(k_0)}] - e_u^{(i)}. \quad (5.37)$$

Рассмотрим проверку

$$A_v = \sum_{i=k_0+1}^{n} \sum_{u=-t}^{m} b_{iu} s_u^{(i)}, \qquad v = 1, 2 \ldots, J, \quad (5.38)$$

являющуюся линейной комбинацией символов синдромов. Эта проверка называется составной проверкой. Поскольку символы $s_u^{(i)}$, задаваемые формулой (5.37), являются линейными комбинациями шумовых символов $e_u^{(i)}$, $i = 1, \ldots, n_0$, возникающих за время $-t \leqslant u \leqslant m$, то составные проверки A_v также являются линейными комбинациями этих символов $e_u^{(i)}$. Принцип порогового декодирования был сформулирован Месси [2] следующим образом.

Определение 5.1. *Система из J составных проверок $\{A_v, v = 1, \ldots, J\}$, контролирующих $e_u^{(i)}$, называется ортогональной относительно $e_u^{(i)}$, если никакой другой шумовой символ, отличный от $e_u^{(i)}$, не контролируется более чем одной проверкой этой системы (шумовой символ контролируется проверкой A_v тогда и только тогда, когда он входит в эту проверку с ненулевым коэффициентом).*

Рассмотрим один из методов мажоритарного декодирования, который позволяет определить $e_u^{(i)}$ из системы J проверок $\{A_v\}$, ортогональных относительно $e_u^{(i)}$. Здесь и далее через $[J]$ будем обозначать наибольшее целое число, не превосходящее J.

Теорема 5.1. *Пусть отличны от нуля не более $[J/2]$ шумовых символов в совокупности $\{e_w^{(j)}\}$, $-t \leqslant w \leqslant m$, $1 \leqslant j \leqslant n_0$, контролируемых системой из J составных проверок $\{A_v\}$, $v = 1, 2, \ldots, J$, ортогональных относительно $e_u^{(i)}$ (т. е. имеется не более $J/2$ ошибок в соответствующих принятых символах). Тогда $e_u^{(i)}$ есть значение, принимаемое большинством составных проверок (в случае, когда ни одно значение из $GF(q)$ не принимается абсолютным большинством составных проверок, считается, что $e_u^{(i)} = 0$).*

Доказательство. Положим $e_u^{(i)} = V$ и допустим, что все другие шумовые символы, входящие в проверки системы $\{A_v\}$, имеют значение 0, Тогда

$$A_v = V, \qquad v = 1, 2, \ldots, J. \quad (5.39)$$

Если $V = 0$, то $[J/2]$ шумовых символов могут изменить значения не более чем $[J/2]$ проверок A_v из (5.39), а поэтому не менее чем $J - [J/2]$ проверок среди $\{A_v\}$ равны нулю. Таким образом, нуль является либо наиболее часто встречающимся (при нечетном J), либо одним из двух одинаково часто встречающихся значений проверок из $\{A_v\}$, однако в любом случае указанное в теореме правило декодирования дает правильное значение шумового символа.

Далее предположим, что $V \neq 0$. В этом случае $e_u^{(i)} = V \neq 0$ и среди других шумовых символов $e_u^{(i)}$ отличными от нуля могут быть самое большее $[J/2] - 1$ символов. Следовательно, не менее чем $J - [J/2] + 1$ проверок A_v в (5.39) равны V, так что указанное в теореме правило декодирования вновь дает правильное значение шумового символа $e_u^{(i)}$.

Если $J = 2t$, то из теоремы 5.1 следует, что символ $e_u^{(i)}$ может быть правильно определен во всех случаях, когда число ошибок не превышает t. Если $J = 2t + 1$, то символ $e_u^{(i)}$ снова может быть правильно определен во всех случаях, когда произошло не более t ошибок; при этом дополнительно могут быть обнаружены любые $t + 1$ ошибок. Это означает, что расстояние Хэмминга между любыми двумя кодовыми словами не меньше чем $J + 1$. Таким образом, имеет место следующее утверждение.

Следствие 5.1. *Если линейный код допускает образование системы из J проверок, ортогональных относительно $e_u^{(i)}$, то минимальное расстояние d_{FD} этого кода не меньше чем $J + 1$, т. е.*

$$d_{FD} \geqslant J + 1. \tag{5.40}$$

Параметр J называется числом ортогональных проверок.

Метод мажоритарного декодирования в случае двоичных кодов сводится к следующему.

Теорема 5.2. *Если $\{A_v\}$ — система из J проверок, ортогональных относительно $e_u^{(i)}$, то алгоритм мажоритарного декодирования таков: выбирать $e_u^{(i)} = 1$ тогда и только тогда, когда сумма всех A_v (рассматриваемых как действительные числа) превосходит порог $J/2$, если J четное, и $(J + 1)/2$, если J нечетное.*

Теорема 5.2 указывает очень простой способ декодирования. А именно если задан код, ортогональный относительно $e_u^{(i)}$, то при приеме следует сначала вычислить синдромы, а затем путем линейного преобразования полученных синдромов найти составные проверки. Далее составные проверки следует подать на входы порогового элемента, имеющего порог $(J/2) + 1$, и испра-

вить возникшие ошибки с помощью выходного символа этого порогового элемента.

Приведенные выше две теоремы касались лишь исправления шумового символа $e_u^{(i)}$, воздействующего на информационный символ в момент u. Однако сверточные коды таковы, что, выполняя в точности те же самые операции только в моменты времени $u+1$, $u+2$, ..., можно провести исправление также и шумовых символов $e_{u+1}^{(i)}$, $e_{u+2}^{(i)}$, ..., действующих в моменты времени $u+1$, $u+2$, ... За начало отсчета времени обычно выбирают момент $u=0$, однако при пороговом декодировании для исправления шумового символа $e_0^{(i)}$ используются символы синдромов от момента 0 до момента m. При этом после нахождения шумового символа иногда с помощью цепи обратной связи «воздействие» $e_0^{(i)}$ на синдром устраняется, а в других случаях цепь обратной связи отсутствует и синдром остается искаженным шумовым символом $e_0^{(i)}$ даже после нахождения последнего. В последних случаях декодирование называют *дефинитным*. Коды, используемые при дефинитном декодировании, имеют приблизительно в 2 раза большее кодовое ограничение, чем коды, используемые при декодировании с обратной связью. При этом, вообще говоря, ухудшаются корректирующие способности кодов, но благодаря отсутствию цепи обратной связи при возникновении шумов, превосходящих допустимый уровень, возникшие ошибки могут распространяться лишь на конечную глубину.

5.5. Самоортогональные коды

5.5.1. Определение самоортогонального кода

При пороговом декодировании сверточных кодов ошибки исправляются последовательно сначала в 1-м принятом блоке длины n_0, затем во 2-м и т. д. При этом корректирующие способности кода определяются тем, как исправляются ошибки в 1-м блоке, так как после исправления ошибок в 1-м блоке исправление ошибок в последующих блоках осуществляется точно так же, как и исправление ошибок в 1-м блоке. Следовательно, при изучении порогового декодирования сверточных кодов можно ограничиться исследованием влияния структуры синдромов и матрицы H на декодирование лишь 1-го блока. Среди кодов, допускающих пороговое декодирование, самыми простыми являются так называемые *самоортогональные коды*.

Определение 5.2. (Месси.) *Сверточный код, для которого при любом* $i=1, 2, \ldots, k_0$ *система проверок, контролирующая*

шумовой символ $e_0^{(i)}$, воздействующий на i-й информационный символ в момент времени 0, ортогональна относительно $e_0^{(i)}$, называется самоортогональным кодом.

Все самоортогональные коды допускают полную ортогонализацию, т. е. с помощью мажоритарного декодирования могут быть исправлены все ошибки, исправление которых гарантируется минимальным расстоянием. Коды этого типа первым исследовал Месси [2]. Метод построения таких кодов с высокой корректирующей способностью, а именно кодов, имеющих при заданном J небольшое кодовое ограничение n_A, был предложен Робинсоном и Бернстейном [3]. Декодирование самоортогональных кодов осуществляется крайне просто, и это связано с тем, что соответствующим образом выбранные символы синдромов ортогональны относительно $e_u^{(i)}$ и при декодировании нет необходимости проводить линейное преобразование символов синдромов для получения составных проверок. Следующая теорема дает необходимые и достаточные условия того, что код является самоортогональным.

Теорема 5.3. (Месси.) *Код является самоортогональным тогда и только тогда, когда совокупность символов синдромов, контролирующих каждый информационный шумовой символ $e_u^{(i)}$, ортогональна относительно $e_u^{(i)}$.*

Доказательство. Очевидно, что если при любом $u \geqslant 1$ совокупность символов синдромов, контролирующих $e_u^{(i)}$, ортогональна относительно $e_u^{(i)}$, то, в частности, это свойство ортогональности выполняется и при $u = 0$, т. е. код является самоортогональным. Докажем обратное утверждение. Предположим, что при некотором $u > 0$ совокупность символов синдромов, контролирующих $e_u^{(i)}$, не является ортогональной относительно $e_u^{(i)}$. Тогда должны существовать два символа синдрома, например $s_t^{(j)}$ и $s_{t'}^{(j')}$, контролирующих один и тот же *информационный шумовой символ* $e_{u'}^{(i')}$ (т. е. шумовой символ, воздействующий на информационный символ), отличный от $e_u^{(i)}$. Из равенства (5.13), определяющего способ построения символа синдрома, непосредственно следует, что $t, t' \geqslant u, u'$. Кроме того, без ограничения общности можно предположить, что $u' > u$. Вновь обращаясь к способу построения символа синдрома, получаем, что символы синдромов $s_{t-u}^{(j)}$ и $s_{t'-u}^{(j')}$ контролируют и $e_{u-u}^{(i)} = e_0^{(i)}$, и $e_{u'-u}^{(i')}$. Следовательно, совокупность символов синдромов, контролирующих $e_0^{(i)}$, не является ортогональной относительно $e_0^{(i)}$, что в свою очередь означает, что код не является самоортогональным.

5.5.2. Простой пример

Прежде чем перейти к методам построения самоортогональных кодов, попытаемся сначала исследовать структуру самоортогональных кодов.

Пример 5.6. Рассмотрим самоортогональный код со скоростью передачи $R = {}^2/_3$ и $J = 3$, задаваемый порождающими многочленами

$$G^{(3)}_{(1)}(D) = 1 + D + D^4,$$
$$G^{(3)}_{(2)}(D) = 1 + D^2 + D^7.$$

Проверочная матрица H^Δ этого кода для интервала времени $-7 \leqslant u \leqslant 7$ имеет следующий вид:

$$H^\Delta = \left[\begin{array}{cccccccccccccccc}
1 & 0 & 0 & 1 & . & 1 & & & 1 & 0 & 0 & 0 & 0 & 1 & 0 & . & 1 & & & & 1 & & \\
 & 1 & 0 & 0 & . & 1 & 1 & & & 1 & 0 & 0 & 0 & 0 & 1 & . & 0 & 1 & & & 1 & & \\
 & & 1 & 0 & . & 0 & 1 & 1 & & & 1 & 0 & 0 & 0 & 0 & . & 1 & 0 & 1 & & & 1 & \\
 & & & 1 & . & 0 & 0 & 1 & 1 & 0 & & 1 & 0 & 0 & 0 & . & 0 & 1 & 0 & 1 & 0 & & 1 \\
 & & & & . & 1 & 0 & 0 & 1 & 1 & & & 1 & 0 & 0 & . & 0 & 0 & 1 & 0 & 1 & & & 1 \\
 & 0 & & & . & 0 & 1 & 0 & 0 & 1 & 1 & & & 1 & 0 & . & 0 & 0 & 0 & 1 & 0 & 1 & & \\
 & & & & . & 0 & 0 & 1 & 0 & 0 & 1 & 1 & & & 1 & . & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \\
 & & & & . & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & & & . & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{array}\right]. \tag{5.41}$$

Можно построить следующие три составные проверки, ортогональные относительно $e^{(1)}_0$:

$$\begin{aligned}
A_1 &= s^{(3)}_0 = e^{(1)}_{-4} + e^{(1)}_{-1} + e^{(1)}_0 + e^{(2)}_{-7} + e^{(2)}_{-2} + e^{(2)}_0 + e^{(3)}_0, \\
A_2 &= s^{(3)}_1 = e^{(1)}_{-3} + e^{(1)}_0 + e^{(1)}_1 + e^{(2)}_{-6} + e^{(2)}_{-1} + e^{(2)}_1 + e^{(3)}_1, \\
A_3 &= s^{(3)}_4 = e^{(1)}_0 + e^{(1)}_3 + e^{(1)}_4 + e^{(2)}_{-3} + e^{(2)}_2 + e^{(2)}_4 + e^{(3)}_4,
\end{aligned} \tag{5.42}$$

и три составные проверки, ортогональные относительно $e^{(2)}_0$:

$$\begin{aligned}
A'_1 &= s^{(3)}_0 = e^{(2)}_{-7} + e^{(2)}_{-2} + e^{(2)}_0 + e^{(1)}_{-4} + e^{(1)}_{-1} + e^{(1)}_0 + e^{(3)}_0, \\
A'_2 &= s^{(3)}_2 = e^{(2)}_{-5} + e^{(2)}_0 + e^{(2)}_2 + e^{(1)}_{-2} + e^{(1)}_{-1} + e^{(1)}_2 + e^{(3)}_2, \\
A'_3 &= s^{(3)}_7 = e^{(2)}_0 + e^{(2)}_5 + e^{(2)}_7 + e^{(1)}_4 + e^{(1)}_7 + e^{(1)}_8 + e^{(3)}_7.
\end{aligned} \tag{5.43}$$

Очевидно, что для этого кода $J = 3$.

Как можно видеть из формул (5.42) и (5.43), символами синдромов, ортогональными относительно $e^{(1)}_0$, являются сим-

волы, порождаемые в моменты времени 0, 1, 4, а символами синдромов, ортогональными относительно $e_0^{(2)}$, являются символы, порождаемые в моменты времени 0, 2, 7. Моменты образования этих символов синдромов соответствуют показателям {0, 1, 4} и {0, 2, 7} степеней неизвестного D с ненулевыми коэффициентами порождающих многочленов $G_{(3)}^{(1)}(D)$ и $G_{(3)}^{(2)}(D)$.

Пусть $\{d_0, d_1, d_2\}$ — показатели степеней неизвестного D с ненулевыми коэффициентами порождающего многочлена. Рассмотрим совокупность

$$\{d_1 - d_0,\ d_2 - d_0,\ d_2 - d_1,\ d_0 - d_1,\ d_0 - d_2,\ d_1 - d_2\}$$

разностей этих показателей степеней. А именно для введенных выше многочленов $G_{(3)}^{(1)}(D)$ и $G_{(3)}^{(2)}(D)$ рассмотрим совокупности чисел {1, 4, 3, —1, —4, —3} и {2, 7, 5, —2, —7, —5}. Совокупность {1, 4, 3, —1, —4, —3} представляет собой совокупность значений индексов u информационных шумовых символов $e_u^{(1)}$, входящих в проверки (5.42) и отличных от $e_0^{(1)}$. Точно так же совокупность {2, 7, 5, —2, —7, —5} совпадает с совокупностью значений индексов u информационных шумовых символов $e_u^{(2)}$, входящих в проверки (5.73) и отличных от $e_0^{(2)}$. Следовательно, необходимым условием ортогональности систем проверок (5.42) и (5.43) относительно соответственно $e_0^{(1)}$ и $e_0^{(2)}$ является отсутствие в каждой из указанных выше совокупностей одинаковых чисел. Кроме того, если обозначить совокупности {0, 1, 4} и {0, 2, 7} через $\{d_0, d_1, d_2\}$ и $\{d_0', d_1', d_2'\}$, то целые числа $d_0 - d_0' = 0$, $d_0 - d_1' = -2$, $d_0 - d_2' = -7$, $d_1 - d_0' = 1$, $d_1 - d_1' = 1$, $d_1 - d_2' = -6$, $d_2 - d_0' = 4$, $d_2 - d_1' = 2$, $d_2 - d_2' = -3$ оказываются значениями индексов u шумовых символов $e_u^{(2)}$, входящих в формулы (5.42). Аналогично целые числа, получающиеся в результате вычитания из каждого числа $\{d_0', d_1', d_2'\}$ каждого числа $\{d_0, d_1, d_2\}$, являются значениями индексов u шумовых символов $e_u^{(1)}$, входящих в (5.43). Следовательно, другим необходимым условием ортогональности систем проверок (5.42) и (5.43) относительно соответственно $e_0^{(1)}$ и $e_0^{(2)}$ является также отсутствие одинаковых чисел и в последних указанных совокупностях. В действительности, как легко видеть, эти условия являются и достаточными, а именно если совокупности целых чисел $\{d_0, d_1, d_2\}$ и $\{d_0', d_1', d_2'\}$ удовлетворяют указанным выше условиям, то задаваемый ими код является самоортогональным.

В общем случае пусть целые числа $d_1 < d_2 < \ldots < d_{q+1}$ являются показателями степеней в порождающем многочлене

$G^{(j)}_{(i)}(D)$, имеющем $q+1$ ненулевых коэффициентов, т. е. пусть

$$G^{(j)}_{(i)}(D) = D^{d_1} + D^{d_2} + \ldots + D^{d_{q+1}}. \tag{5.44}$$

Множество $\{d_i - d_j | i > j\}$ из $q(q+1)/2$ разностей целых чисел d_i, $i = 1, 2, \ldots, q+1$, будем называть совокупностью разностей порождающего многочлена $G^{(j)}_{(i)}$.

Теорема 5.4. (Маси и Робинсон.) *Сверточные коды с $k_0 = 1$ или $k_0 = n_0 - 1$ являются самоортогональными тогда и только тогда, когда совокупности разностей их $n_0 - 1$ порождающих многочленов не пересекаются, а кроме того, ни одна из них не содержит одно и то же число дважды.*

Доказательство. Рассмотрим случай $k_0 = n_0 - 1$, когда существует только один синдром $S^{(n_3)}(D)$. Пусть $d_1 < d_2 < \ldots < d_Y$ и $d'_1 < d'_2 < \ldots < d'_{Y'}$ — показатели степеней неизвестного с ненулевыми коэффициентами порождающих многочленов $G^{(j)}_{(n_3)}(D)$ и $G^{(j')}_{(n_3)}(D)$. Согласно формулам (5.11) и (5.13), символами синдрома, контролирующими шумовой символ $e^{(j)}_u$, являются символы, появляющиеся в моменты времени $t = u + d_1$, $u + d_2$, ..., $u + d_Y$. Индексы символов j-й и j'-й информационных шумовых последовательностей, контролируемых этими символами синдромов, могут быть найдены с помощью следующей таблицы:

Символ синдрома	Индексы контролируемых символов j-й информационной шумовой последовательности	Индексы контролируемых символов j'-й информационной последовательности
$s^{(n_0)}_{u+d_1}$	$u + d_1 - d_Y$, $u + d_1 - d_{Y-1}$, ..., $u + d_1 - d_2$, u	$u + d_1 - d'_{Y'}$, ..., $u + d_1 - d'_2$, $u + d_1 - d'_1$
$s^{(n_0)}_{u+d_2}$	$u + d_2 - d_Y$, $u + d_2 - d_{Y-1}$, u, $u + d_2 - d_1$	$u + d_2 - d'_{Y'}$, ..., $u + d_2 - d'_1$
$\vdots$	$\vdots$	
$s^{(n_0)}_{n+d_Y}$	u, $u + d_Y - d_{Y-1}$, ... $u + d_Y - d_1$	$u + d_Y - d'_{Y'}$, ..., $u + d_Y - d'_1$

Эти символы синдромов ортогональны относительно $e^{(i)}_u$ тогда и только тогда, когда выполняются следующие два условия:

1) Никакой индекс, отличный от u, не входит сразу в две строки второго столбца таблицы. Действительно, если при некоторых i, j, k, l, $i > k$ имеет место равенство $u + d_i - d_j = u + d_k - d_l$, то $d_i - d_j = d_k - d_l$, а это означает, что некоторое число входит в совокупность разностей $G^{(j)}_{(n_0)}(D)$ по крайней мере дважды.

2) Никакой индекс не входит сразу в две строки третьего столбца таблицы. Действительно, если при некоторых i, j', k, l', $i > k$ имеет место равенство $u + d_i - d_{j'} = u + d_k - d_{l'}$, то $d_i - d_k = d_{l'} - d_{j'}$ и совокупности разностей $G^{(j')}_{(n_0)}(D)$ и $G^{(j)}_{(n_0)}(D)$ содержат одно и то же целое число.

Это доказывает теорему в случае $k_0 = n_0 - 1$. Аналогично теорема доказывается и в случае $k_0 \neq n_0 - 1$.

Из этой теоремы непосредственно следует, что после определения $n_0 - 1$ порождающих многочленов, для которых все совокупности разностей состоят из различных чисел и не пересекаются, сразу получаем самоортогональный код.

Существует бесконечно много совокупностей целых чисел $\{d_1, d_2, \ldots, d_{q+1}\}$, множества разностей $\{d_i - d_j | i > j\}$ которых состоят из различных целых чисел. Однако для построения кодов с наивысшими корректирующими способностями, а именно для достижения минимально возможного кодового ограничения при заданном J, необходимы такие совокупности целых чисел, для которых максимальное целое число, входящее в множество разностей $\{d_i - d_j | i > j\}$, имело бы минимально возможное значение.

Простое совершенное разностное множество порядка q представляет собой совокупность $q+1$ целых чисел $(d_1, d_2, \ldots, d_{q+1})$, каждая из $q^2 + q$ разностей $d_i - d_j$, $i \neq j$, $i, j = 1, 2, \ldots, q+1$, которой сравнима по модулю $q^2 + q + 1$ с одним из чисел $1, 2, \ldots, q^2 + q$. Сингер [37] доказал существование простых совершенных разностных множеств любого порядка q, являющегося степенью простого числа, и более того, указал метод их построения. Например, как показано ниже, при $q = 3$ совокупность целых чисел $\{0, 1, 3, 9\}$ является простым совершенным разностным множеством:

$$1 - 0 \equiv 1 \quad 0 - 9 \equiv 4 \quad 3 - 9 \equiv 7, \quad 0 - 3 \equiv 10$$

$$3 - 1 \equiv 2 \quad 1 - 9 \equiv 5 \quad 9 - 1 \equiv 8, \quad 1 - 3 \equiv 11 \pmod{13}.$$

$$3 - 0 \equiv 3 \quad 9 - 3 \equiv 6 \quad 9 - 0 \equiv 9, \quad 0 - 1 \equiv 12$$

5.5.3. Коды, строящиеся с помощью простых совершенных разностных множеств

Если задано простое совершенное разностное множество, то теорема 5.4 позволяет сразу построить самоортогональный код со скоростью $R = 1/2$ и $J = q + 1$. Для этого достаточно в качестве порождающего многочлена этого кода взять многочлен

$$G^{(2)}_{(1)}(D) = D^{d_1} + D^{d_2} + \ldots + D^{d_{q+1}}. \tag{5.45}$$

Коды со скоростью $R = (n_0 - 1)/n_0$ или $1/n_0$ могут быть построены с помощью следующей процедуры.

Если из простого совершенного разностного множества $\{d_1, d_2, \ldots, d_{q+1}\}$ выделить подмножество $\{d'_1, d'_2, \ldots, d'_r\}$, $r \leqslant q+1$, из r следующих друг за другом целых чисел и построить совокупность $\{d'_i - d'_j \mid i > j\}$ из $r(r-1)/2$ разностей этих чисел, то все эти разности будут различными целыми числами. Это означает, что указанное выше подмножество само является простым совершенным разностным множеством. Следовательно, разбивая простое совершенное разностное множество на $n_0 - 1$ непересекающихся подмножеств следующих подряд целых чисел и рассматривая эти подмножества как совокупности показателей степеней неизвестного с ненулевыми коэффициентами $n_0 - 1$ порождающих многочленов, можно построить самоортогональный код со скоростью $R = (n_0 - 1)/n_0$ или $1/n_0$. Если задано простое совершенное разностное множество, то, вычитая из каждого элемента этого множества минимальный элемент, опять получаем разностное множество. Пользуясь этим свойством разностных множеств, можно понизить степени порождающих многочленов, а следовательно, и объем памяти, используемый кодером и декодером.

Пример 5.7. Простое совершенное разностное множество $\{0, 1, 3, 9\}$ с $q = 3$ позволяет построить самоортогональный код со скоростью $R = 1/2$ и $J = 4$. Для этого достаточно взять порождающий многочлен

$$G^{(1)}_{(2)}(D) = 1 + D + D^3 + D^9. \tag{5.46}$$

Разбивая это разностное множество на два подмножества $\{0, 1\}$ и $\{3, 9\}$ и вычитая из элементов последнего минимальный элемент 3, получим подмножества $\{0, 1\}$ и $\{0, 6\}$. С помощью этих подмножеств можно построить код со скоростью $R = 1/3$ и $J = 4$, положив

$$\begin{aligned} G^{(1)}_{(2)}(D) &= 1 + D, \\ G^{(1)}_{(3)}(D) &= 1 + D^6, \end{aligned} \tag{5.47}$$

и код со скоростью $R = {}^2/_3$ и $J = 2$, положив

$$G^{(1)}_{(3)}(D) = 1 + D,$$
$$G^{(2)}_{(3)}(D) = 1 + D^6. \tag{5.48}$$

Таким образом, с помощью простого совершенного разностного множества всегда можно построить самоортогональный код со скоростью $R = (n_0 - 1)/n_0$ и $1/n_0$. Однако при построении необходимо стремиться уменьшить кодовое ограничение при фиксированном J, поскольку это повышает корректирующие способности кода и позволяет уменьшить объем памяти кодера и декодера. Учитывая это, оптимальным кодом будем называть код, который при заданных скорости передачи R, числе ортогональных проверок J и длине блока n_0 имеет минимально возможное кодовое ограничение $n_A = (m + 1)/n_0$. Робинсон и Бернстейн предложили следующую эффективную процедуру оптимизации построения самоортогональных кодов на основе совершенных разностных множеств [3]:

1) Строится следующее разбиение множества разностей простого совершенного разностного множества порядка q на q подмножеств. Первое подмножество:

$$a_{1,\,i} = d_{i+1} - d_i, \quad i = 1, 2, \ldots, q,$$
$$a_{1,\,q+1} = d_1 - d_{q+1} + q^2 + q + 1.$$

Остальные $q - 1$ подмножеств разбиения $\{a_{i,1}, \ldots, a_{i,\,q+1}\}$, $2 \leqslant j \leqslant q$, определяются равенством

$$a_{i,\,j} = \sum_{k=j}^{i+j-1} a_{1,\,k}, \tag{5.49}$$

где $a_{1,\,k} = a_{1,\,k-q-1}$, $k > q + 1$, $1 \leqslant i \leqslant q$, $1 \leqslant j \leqslant q + 1$. Например, с помощью простого совершенного разностного множества (0, 1, 3, 9) получается следующее разбиение множества разностей:

$$\begin{matrix} 1, & 2, & 6, & 4 \\ 3, & 8, & 10, & 5 \\ 9, & 12, & 11, & 7 \end{matrix} \tag{5.50}$$

Каждый элемент этого разбиения равен по модулю $q^2 + q + 1$ одной из разностей двух элементов простого совершенного разностного множества:

$$a_{l,\,j} = d_r - d_j = \sum_{k=1}^{i+j-1} a_{1,\,k} - \sum_{k=j-1}^{i+j-1} a_{1,\,k} \pmod{q^2 + q + 1}, \tag{5.51}$$

$$r \equiv i + j \pmod{q + 1}. \tag{5.52}$$

Например, в (5.50)

$$a_{23} = 10 \equiv d_1 - d_3 = 1 - 4 \pmod{13}.$$

В силу основного свойства простого совершенного разностного множества каждое из целых чисел от 1 до $q^2 + q$ входит в это разложение ровно один раз.

2) Рассмотрим множество $\{a_{0j} = 0, a_{1j}, \ldots, a_{qj}\}$, $a_{0j}=0 < a_{1j} < \ldots < a_{qj}$. Из формул (5.51) и (5.52) получаем

$$a_{i+k,\,j} - a_{i,\,j} \equiv (d_{r'} - d_j) - (d_r - d_j) = d_{r'} - d_r \pmod{q^2 + q + 1}, \tag{5.53}$$

$$r' \equiv i + k + j \pmod{q + 1},$$

$$r \equiv i + j \pmod{q + 1},$$

$$i = 0, \ldots, q, \quad k = 1, \ldots, q.$$

Из этих соотношений видно, что совокупность разностей чисел из рассматриваемого множества совпадает с совокупностью $q^2 + q$ разностей $d_i - d_j$, $i \neq j$, $i, j = 1, 2, \ldots, q + 1$, чисел простого совершенного разностного множества $\{d_1, \ldots, d_{q+1}\}$. Следовательно, множество $\{a_{0j} = 0, a_{1j}, \ldots, a_{qj}\}$ также является простым совершенным разностным множеством. Например, множество целых чисел, образующих 2-й столбец разложения (5.50), таково, что

$$\begin{array}{llll} 0-12\equiv 1 & 12-8\equiv 4 & 2-8\equiv 7 & 12-2\equiv 10 \\ 2-0\equiv 2 & 0-8\equiv 5 & 8-0\equiv 8 & 0-2\equiv 11 \\ 2-12\equiv 3 & 8-2\equiv 6 & 8-12\equiv 9 & 12-0\equiv 12 \end{array} \pmod{13},$$

и является простым совершенным разностным множеством. Таким образом, если задано одно совершенное разностное множество, то можно с его помощью построить $q + 1$ совершенных разностных множеств $\{0, a_{1j}, a_{2j}, \ldots, a_{qj}\}$, $j = 1, \ldots, q + 1$.

3) Если из построенных таким образом $q + 1$ простых совершенных разностных множеств

$$\begin{aligned} &(0, \ a_{11}, \ldots, a_{q1}) = (0, \ d_1, \ldots, d_q), \\ &(0, \ a_{12}, \ldots, a_{q2}), \\ &\quad\vdots \\ &(0, \ a_{1,\,q+1}, \ldots, a_{q,\,q+1}) \end{aligned}$$

выбрать множество, максимальное из чисел которого является наименьшим из максимальных чисел $a_{q1}, a_{q2}, \ldots, a_{q,\,q+1}$ указанных выше множеств, то можно построить порождающий многочлен кода минимальной степени.

Например, разложение (5.50) позволяет построить следующие четыре порождающих многочлена кодов со скоростью $R = 1/2$ и $J = 4$:

$$G_{(2)}^{(1)}(D) = 1 + D + D^3 + D^9,$$
$$G_{(2)}^{(1)}(D) = 1 + D^2 + D^8 + D^{12},$$
$$G_{(2)}^{(1)}(D) = 1 + D^6 + D^{10} + D^{11},$$
$$G_{(2)}^{(1)}(D) = 1 + D^4 + D^5 + D^7.$$

Из этих многочленов следует выбрать многочлен

$$G_{(2)}^{(1)}(D) = 1 + D^4 + D^5 + D^7,$$

имеющий минимальную степень.

5.5.4. Примеры кодов

Робинсон и Бернстейн, используя описанную выше процедуру понижения степеней порождающих многочленов, методом перебора с помощью простых совершенных разностных множеств, порядки которых являются степенями простых чисел, не превосходящими 30, нашли порождающие многочлены кодов со скоростями $R = (n_0 - 1)/n_0$ и $R_0 = 1/n_0$, имеющие минимальные степени. Полученные ими результаты приведены в табл. 5.1.

5.5.5. Оптимальность кодов

Один из путей определения оптимальности построенных указанным выше способом кодов заключается в том, чтобы найти нижнюю границу кодового ограничения и далее сравнивать с ней кодовые ограничения построенных кодов. Уэлдон [40] получил следующие нижние границы кодового ограничения.

Для любого кода со скоростью передачи $R = (n_0 - 1)/n_0$ и J ортогональными проверками $(n_0 - 1)$ совокупностей разностей порождающих многочленов из различных целых чисел и каждая из этих совокупностей разностей содержит по крайней мере $J(J-1)/2$ элементов. Следовательно, максимальное число, входящее в эти $(n_0 - 1)$ совокупностей разностей, по крайней мере не меньше чем $(n_0 - 1)J(J-1)/2$. Добавляя к этому числу единицу и умножая полученную сумму на n_0, получаем следующую нижнюю границу для кодового ограничения:

$$n_A \geqslant n_0\left[\frac{(n_0 - 1)J(J-1)}{2} + 1\right]. \qquad (5.54)$$

Рассмотрим самоортогональные коды со скоростью $R = 1/n_0$. Поскольку для каждого из этих кодов могут быть построены

Таблица 5.1

Самоортогональные коды

$R=\frac{1}{J}=1/2$	n_A	Ненулевые коэффициенты порождающих многочленов, $n_0=2$
4	14	0, 2, 5, 6
6	36	0, 2, 7, 13, 16, 17
8	72	0, 7, 10, 16, 18, 30, 31, 35
10	112	0, 2, 14, 21, 29, 32, 45, 49, 54, 55
12	172	0, 2, 6, 24, 29, 40, 43, 55, 68, 75, 76, 85
14	256	0, 5, 28, 38, 41, 49, 50, 68, 75, 92, 107, 121, 123, 127
16	360	0, 6, 19, 40, 58, 67, 78, 83, 109 132, 133, 162, 165, 169, 177, 179
18	434	0, 2, 10, 22, 53, 56, 82, 83, 89, 98, 130, 148, 153, 167, 188, 192. 205, 216
20	568	0, 24, 30, 43, 55, 71, 75, 89, 104, 125, 127, 162, 167, 189, 206, 215, 272, 275, 282, 283
22	718	0, 3, 16, 45, 50, 51, 65, 104, 125, 142, 182, 206, 210, 218, 228, 237, 289, 300, 326, 333, 356, 358
24	852	0, 22, 41, 57, 72, 93, 99, 139, 147, 173, 217. 220, 234, 273, 283, 285, 296, 303, 328, 387, 388, 392, 416, 425

$R=\frac{2}{J}=2/3$	$R=\frac{1}{J}=1/3$	n_A	Ненулевые коэффициенты порождающих многочленов, $n_0=3$
2	4	9	0, 1 0, 2
	6	27	0, 1, 7 0, 3, 8
4	8	42	0, 8, 9, 12 0, 6, 11, 13
	10	69	0, 2, 9, 21, 22 0, 4, 10, 15, 18
6	12	120	0, 14, 21, 23, 36, 39 0, 8, 27, 28, 32, 38
	14	165	0, 1, 28, 31, 44, 46, 52 0, 7, 17, 29, 40, 49, 54
8	16	237	0, 19, 21, 45, 63, 73, 74, 78 0, 27, 39, 47, 61, 64, 70, 77
	18	309	0, 16, 37, 44, 47, 70, 82, 83, 97 0, 22, 24, 30, 41, 73, 93, 98, 102
10	20	393	0, 1, 6, 25, 32, 72, 100, 108, 120, 130 0, 23, 39, 57, 60, 74, 101, 103, 112, 116
	22	507	0, 5, 12, 53, 72, 81, 135, 136, 146, 159, 167 0, 17, 46, 50, 52, 66, 88, 125, 150, 165, 168
12	24	588	0, 17, 46, 50, 52, 66, 88, 125, 150, 165, 168, 195 0, 26, 34, 47, 57, 58, 112, 121, 140, 181, 188, 193

Продолжение табл. 5.1

$R=2/3$ J	$R=1/3$ J	n_A	Ненулевые коэффициенты порождающих многочленов, $n_0=3$
	26	753	0, 16, 64, 104, 122, 124, 146, 159, 167, 171, 178, 237, 240 0, 30, 39, 53, 68, 129, 139, 165, 170, 216, 222, 249, 250
14	28	867	0, 2, 7, 42, 45, 117, 163, 185, 195, 216, 229, 246 255, 279 0, 8, 12, 27, 28, 64, 113, 131, 154, 160, 208, 219, 233, 288

$R=3/4$ J	$R=1/4$ J	n_A	Ненулевые коэффициенты порождающих многочленов, $n_0=4$
4	12	80	0, 3, 15, 19 0, 8, 17, 18 0, 6, 11, 13
6	18	248	0, 12, 16, 42, 48, 61 0, 3, 5, 14, 34, 58 0, 15, 22, 23, 40, 50
8	24	512	0 16, 31, 34, 48, 75, 85, 124 0, 28, 35, 47, 58, 71, 80, 100 0, 21, 25, 26, 81, 87, 89, 127
10	30	812	0, 7, 27, 76, 113, 137, 155, 156, 170, 202 0, 8, 38, 48, 59, 82, 111, 146, 150, 152 0, 12, 25, 28, 78, 83, 100, 109, 145, 199

$R=4/5$ J	$R=1/5$ J	n_A	Ненулевые коэффициенты порождающих многочленов, $n_0=5$
4	16	135	0, 16, 20, 21 0, 2, 10, 25 0, 14, 17, 26 0, 11, 18, 24
6	24	415	0, 5, 26, 51, 55, 69 0, 6, 7, 41, 60, 72 0, 8, 11, 24, 44, 82 0, 10, 32, 47, 49, 77
8	32	895	0, 19, 59, 68, 85, 88, 103, 141 0, 39, 87, 117, 138, 148, 154, 162 0, 2, 13, 25, 96, 118, 168, 172 0, 7, 65, 70, 97, 98, 144, 178

$(n_0-1)J$ ортогональных проверок для одной информационной последовательности, то число ортогональных проверок кода по меньшей мере равно $J'=(n_0-1)J$. Так как кодовое ограничение одно и то же, то, подставляя J из последнего равенства и опуская для простоты штрих, получаем следующую нижнюю границу:

$$n_A \geqslant n_0\left[\frac{J}{2}\left(\frac{J}{n_0-1}-1\right)+1\right]. \tag{5.55}$$

5.6. Ортогонализируемые коды

Ортогонализируемые коды были введены Месси как коды, допускающие пороговое декодирование [2]. По сравнению с самоортогональными кодами эти коды могут иметь при заданном минимальном расстоянии (или числе ортогональных проверок) меньшее кодовое ограничение и в этом смысле обладают более высокими корректирующими способностями. Однако эти коды имеют и ряд недостатков, в частности они строятся методом перебора и, как указывается ниже, могут иметь бесконечную глубину распространения ошибок.

Ортогонализируемый код с минимальным расстоянием d (или $J=d-1$ ортогональными проверками) — это код, который допускает построение для каждого информационного символа, входящего в декодируемый блок, J составных проверок, являющихся линейными комбинациями символов синдромов.

5.6.1. Простейшие примеры

Примером ортогонализируемого кода является код, рассмотренный в начале разд. 5.3 и имеющий скорость $R=1/2$ и кодовое ограничение $n_A=12$. Рассмотрим этот код более детально. При пороговом декодировании с обратной связью этого кода посредством линейного преобразования символов синдрома $s_0^{(2)}$, $s_1^{(2)}$, ..., $s_5^{(2)}$, формирующихся в моменты времени от 0 до 5, могут быть построены составные проверки:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_0^{(2)} \\ s_1^{(2)} \\ s_2^{(2)} \\ s_3^{(2)} \\ s_4^{(2)} \\ s_5^{(2)} \end{bmatrix}. \tag{5.56}$$

Заметим, что следующее линейное преобразование синдрома [который определяется в данном случае соотношением (5.34)]:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_0^{(2)} \\ s_1^{(2)} \\ s_2^{(2)} \\ s_3^{(2)} \\ s_4^{(2)} \\ s_5^{(2)} \end{bmatrix} \tag{5.57}$$

также позволяет построить систему составных проверок, ортогональных относительно $e_0^{(1)}$.

Линейное преобразование, с помощью которого по синдрому находятся составные проверки, при декодировании с обратной связью обычно задается с помощью матрицы размера $J \times (m+1)$. Каждый столбец этой матрицы линейного преобразования содержит не более одного ненулевого элемента, а для самоортогональных кодов не более чем один ненулевой элемент входит также и в каждую строку этой матрицы.

В случае самоортогональных кодов матрица линейного преобразования определяется однозначно построением структуры синдрома. В то же время, как показывает вышеприведенный пример, для других ортогонализируемых кодов матрица линейного преобразования может определяться и неоднозначно. Следовательно, при задании ортогонализируемого кода необходимо указывать не только порождающие многочлены, но и линейное преобразование, с помощью которого по синдрому вычисляются составные проверки. Поскольку последнее преобразование заключается в вычислении некоторых сумм строк проверочной матрицы, то преобразование (5.56) можно задать следующим образом: 0^2, 3^2, 4^2, 5^21^2, где основные цифры указывают момент формирования символа синдрома, а верхние индексы — синдром, к которому они принадлежат. Символы синдромов, участвующие в формировании различных проверок, разделены запятыми.

5.6.2. Способ построения кодов

Построение ортогонализируемых кодов и линейных преобразований для их ортогонализации осуществляется полностью методом перебора. Исключение составляют лишь коды со скоро-

стью $R = 1/2$, при построении которых можно воспользоваться следующим приемом:

1) Пусть элемент проверочного треугольника, расположенный в 1-й строке и 1-м столбце, равен 1, а элементы, расположенные в 1-м столбце и в строках со второй по $(n_A/4)$-ю, равны 0. Тогда шумовые символы с $e^{(1)}_{n_A/4}$ по $e^{(1)}_{(n_A/2)-1}$ могут иметь отличные от нуля коэффициенты только в одной строке треугольника.

2) Каждая из строк проверочного треугольника со второй по $(n_A/4)$-ю может быть использована для исключения одного ненулевого коэффициента при одном из шумовых символов $e^{(1)}_1, \ldots, e^{(1)}_{(n_A/4)-1}$. Например, в (5.56) сложение $s^{(2)}_1$ с $s^{(2)}_5$ позволяет исключить шумовой символ $e^{(1)}_1$, входящий в $s^{(2)}_5$. Заметим, что первая строка проверочного треугольника H^{Δ} дает проверку, контролирующую $e^{(1)}_0$ и $e^{(2)}_0$. Таким образом, задача построения кодов со скоростью $R = 1/2$ свелась к выбору последних $n_A/4$ строк проверочного треугольника H^{Δ}, таких, что из них можно составить $J - 1$ проверок, контролирующих $e^{(1)}_0$ и обладающих тем свойством, что ни один из шумовых символов $e^{(1)}_1, \ldots, e^{(1)}_{(n_A/4)-1}$ не контролируется трижды.

5.6.3. Примеры кодов

В табл. 5.2 приведены ортогонализируемые коды, построенные Месси [2] описанным выше способом.

Рассматриваемые в этом разделе ортогонализируемые коды и самоортогональные коды из предыдущего раздела являются основными сверточными кодами, исправляющими случайные ошибки и допускающими пороговое декодирование. Поскольку, как уже указывалось выше, пороговое декодирование сверточных кодов в принципе аналогично мажоритарному декодированию блоковых кодов, то попробуем установить взаимосвязь между указанными выше кодами и блоковыми кодами, допускающими мажоритарное декодирование.

5.6.4. Связь с блоковыми кодами

Одним из больших классов блоковых кодов, допускающих пороговое декодирование, является класс обобщенных кодов Рида — Маллера, построенный Уэлдоном [44]. Следует заметить, что если обобщенные коды Рида — Маллера, вообще говоря,

являются кодами, ортогонализируемыми за L шагов, то сверточные коды, как самоортогональные, так и ортогонализируемые, являются кодами, ортогонализируемыми за один шаг. В частно-

Таблица 5.2

Ортогонализируемые коды [22]

R	l	^nA	Код [1]	Правило ортогонализации [2]
$1/2$	2	4	$(0, 1)^2$	$0^2, 1^2$
	4	12	$(0, 3, 4, 5)^2$	$0^2, 3^2, 4^2, 1^25^2$
	6	24	$(0, 6, 7, 9, 10, 11)^2$	$0^2, 6^2, 7^2, 9^2, 1^23^210^2, 4^28^211^2$
	8	44	$(0, 11, 13, 16, 17, 19, 20, 21)^2$	$0^2, 11^2, 13^2, 16^2, 17^2, 2^23^26^219^2, 4^214^220^2, 1^25^28^215^221^2$
	10	72	$(0, 18, 19, 27, 28, 29, 30, 32, 33, 35)^2$	$0^2, 18^2, 19^2, 27^2, 1^29^228^2, 10^22^229^2, 11^230^231^2, 13^221^223^232^2, 14^233^234^2, 2^23^216^224^226^235^2$
	12	104	$(0, 26, 27, 39, 40, 41, 42, 44, 45, 47, 48, 51)^2$	$0^2, 26^2, 27^2, 39^2, 1^213^240^2, 14^228^241^2, 15^242^243^2, 17^229^231^244^2, 18^245^246^2, 2^23^220^232^234^247^2, 21^235^248^249^250^2, 24^230^233^236^238^251^2$
$1/3$	4	9	$(0, 1)^2$ $(0, 2)^3$	$0^2, 0^3, 1^2, 2^3$
	6	15	$(0, 1)^2$ $(0, 2, 3, 4)^3$	$0^2, 0^3, 1^2, 2^3, 1^23^3, 2^24^3$
	8	24	$(0, 1, 7)^2$ $(0, 2, 3, 4, 6)^2$	$0^2, 0^3, 1^2, 2^3, 1^33^3, 2^24^3, 7^2, 3^25^26^26^3$
	10	33	$(0, 1, 9)^2$ $(0, 1, 2, 3, 5, 8, 9)^3$	$0^2, 0^3, 1^2, 2^22^3, 9^2, 3^34^3, 3^25^25^3, 1^34^26^26^3, 8^28^3, 7^39^310^3$
$1/5$	4	5	$(0)^2$ $(0)^3$ $(0)^4$ $(0)^5$	$0^2, 0^3, 0^4, 0^5$
	6	10	$(1)^2$ $(1)^3$	$1^21^4, 1^31^5$
	8	15	$(2)^2$ $(2)^4$	$2^22^3, 2^12^5$
	10	20	$(3)^2$ $(3)^5$	$3^5, 3^23^3$

[1]) Запись $(i_1, i_2, \ldots, i_k)^2 \ldots (j_1, j_2, \ldots, j_m)^n$ означает, что ненулевыми коэффициентами порождающих многочленов кода со скоростью $1/n$ являются $g^{(2)}_{i_1}, g^{(2)}_{i_2}, \ldots, g^{(2)}_{i_k}$, $g^{(n)}_{j_1}, g^{(n)}_{j_2}, \ldots, g^{(n)}_{j_m}$. — *Прим. перев.*

[2]) Используемая в таблице запись правил ортогонализации поясняется в конце разд. 5.6.1. — *Прим. перев.*

сти, Месси [2] показал, как описано ниже, что если систематический сверточный код со скоростью передачи $R = 1/n_0$ является кодом, ортогонализируемым за L шагов, то он допускает также ортогонализацию и за один шаг.

Действительно, проверочный треугольник H^{Δ} сверточного кода со скоростью $R = 1/n_0$ имеет следующий вид:

$$H^{\Delta} = \begin{bmatrix} g^{(1)}_{(2),\,0} & & & \\ g^{(1)}_{(2),\,1} & g^{(1)}_{(2),\,0} & & \\ \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \\ \cdot & & \cdot & \\ g^{(1)}_{(2),\,m} & \cdots\cdots & g^{(1)}_{(2),\,0} & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ g^{(1)}_{(n_0),\,0} & & & \\ g^{(1)}_{(n_0),\,1} & g^{(1)}_{(n_0),\,0} & & \\ \cdot & \cdot & & \\ \cdot & \cdot & \cdot & \\ \cdot & & \cdot & \\ g^{(1)}_{(n_0),\,m} & \cdots\cdots & g^{(1)}_{(n_0),\,0} & \end{bmatrix}. \qquad (5.58)$$

Предположим, что существует система J проверок, ортогональных относительно некоторой суммы информационных шумовых символов $e^{(1)}_{a_1} + e^{(1)}_{a_2} + \ldots + e^{(1)}_{a_j}$. Это означает, что существует J подмножеств строк матрицы H^{Δ}, сумма строк каждого из которых имеет символы 1 на позициях, соответствующих информационным шумовым символам $e^{(1)}_{a_1}, e^{(1)}_{a_2}, \ldots, a^{(1)}_{a_j}$ и обладающих тем свойством, что никакие две суммы строк этих подмножеств не содержат символ 1 на одной и той же позиции, отличной от указанных выше. Другими словами, J составных проверок, порождаемых этими подмножествами, контролируют символы $e^{(1)}_{a_1}, \ldots, e^{(1)}_{a_j}$, но никакой другой шумовой символ не контролируется ими дважды. Предположим, что $a_1 < a_2 < \ldots < a_j$. Рассмотрим матрицу, полученную из H^{Δ} удалением всех столбцов слева вплоть до столбца, соответствующего $e^{(1)}_{a_j}$. Строки этой усеченной матрицы позволяют построить J проверок, ортогональных относительно $e^{(1)}_{a_j}$. Из симметрии проверочного треугольника следует, что расположение коэффициентов в этой усеченной матрице такое же, как и в матрице, получающейся путем отбрасывания нижних строк проверочных треугольников в количестве, равном числу отброшенных столбцов. Следовательно, из строк матрицы H^{Δ} можно построить J проверок, ортогональных относительно $e^{(1)}_0$. Другими словами, число проверок, ортогональных относительно $e^{(1)}_0$, не меньше, чем число проверок, ортогональных относительно любой суммы шумовых информационных символов.

Это означает, что если некоторый код допускает ортогонализацию за L шагов, то он ортогонализуем и в один шаг, а следовательно, сверточные коды со скоростью $R = 1/n_0$, ортогонализируемые за L шагов, рассматривать не имеет смысла.

5.6.5. Сопоставление с циклическими кодами, построенными на основе разностных множеств

При первой же попытке установить взаимосвязь между циклическими разностными кодами и самоортогональными кодами сразу можно увидеть, что и те и другие строятся на основе простых совершенных разностных множеств и допускают ортогонализацию за один шаг. Однако эти коды имеют и принципиальные различия. В частности, свойство цикличности разностных циклических кодов требует, чтобы число m, определяющее модуль $m^2 + m + 1$ простого совершенного разностного множества, было степенью числа 2, так что длина n этих кодов, равная модулю, имеет вид $n = 2^{2s} + 2^s + 1$, где s — целое положительное число. В то же время кодовое ограничение самоортогонального кода определяется максимальным элементом простого совершенного разностного множества, на основе которого строится код, и непосредственно не связано с модулем $m^2 + m + 1$ этого разностного множества. Далее, если при построении самоортогональных кодов простые совершенные разностные множества используются для удобства оптимизации характеристик кодов, то при построении разностных циклических кодов простые совершенные разностные множества обеспечивают цикличность кодов, являющуюся одним из важнейших свойств последних. В табл. 5.3 приведены для сравнения параметры самоортогональных кодов со скоростью $R = 1/2$ и разностных циклических кодов, имеющих одно и то же значение $d = J + 1$ и строящихся на основе одного и того же простого совершенного разностного множества или двух различных простых совершенных разностных множеств, имеющих один и тот же максимальный элемент. Простые совершенные разностные множества, использующиеся для построения самоортогональных кодов со скоростью $R = 1/2$, при фиксированном J приводят к эквивалентным кодам, если только они имеют одинаковые максимальные элементы. Следовательно, самоортогональные коды со скоростью $R = 1/2$ и разностные циклические коды, имеющие одинаковое расстояние $d = J + 1$, могут быть построены на основе одного и того же простого совершенного разностного множества.

Самоортогональные коды и разностные циклические коды строятся на основе одних и тех же разностных множеств, однако методы их построения совершенно различны. В то же время

Таблица 5.3

Самоортогональные коды и разностные циклические коды (обозначаются соответственно через *S* и *D*)

d	Код	n или n_A	R	Простое или совершенное разностное множество
4	*D*	7	3/7	0, 2, 3
	S	8	4/8	0, 2, 3
6	*D*	21	11/21	0, 2, 7, 8, 11
	S	24	12/24	0, 2, 7, 10, 11
10	*D*	73	45/73	0, 2, 10, 24, 25, 29, 36, 42, 45
	S	92	46/92	0, 3, 9, 16, 20, 21, 35, 43, 45
18	*D*	273	191/273	0, 18, 24, 46, 50, 67, 103, 112, 115, 126, 128, 159, 166, 167, 186, 196, 201
	S	404	202/404	0, 18, 24, 46, 50, 67, 103, 112 115, 126, 128, 159, 166, 167, 186, 196, 201

самоортогональные коды и квазициклические самоортогональные коды, предложенные Таунсендом и Уэлдоном [40], очень похожи друг на друга. Так, квазициклический самоортогональный код можно построить по самоортогональному коду следующим образом. Входные последовательности $I^{(j)}(D)$ ограничим по модулю степени $M-1$, а именно будем рассматривать лишь Mk_0 информационных символов. Проверочную последовательность $\sum_{j=0}^{k_0} G_{(i)}^{(j)}(D)\, I^{(j)}(D)$ также ограничим по модулю степени $M-1$, полагая $D^M=1$. Полученный таким образом блоковый код длины Mn_0 с Mk_0 информационными символами и скоростью $R=k_0/n_0$ является квазициклическим самоортогональным кодом. Однако в случае квазициклических кодов модуль M должен выбираться таким образом, чтобы совокупности разностей разностного множества не пересекались. Поэтому разностные множества, используемые для построения самоортогональных кодов, и разностные множества, используемые для построения квазициклических самоортогональных кодов с той же скоростью передачи R и тем же числом ортогональных проверок J, часто оказываются различными.

5.7. Распространение ошибок

При декодировании сверточных кодов часто наблюдается явление распространения ошибок, которое состоит в том, что однажды возникшая ошибка декодирования может привести к

возникновению ошибок в последующие моменты времени. В случае блоковых кодов, когда информационные символы разбиваются на блоки и каждый блок кодируется независимо от других, возникшая ошибка декодирования может оказать влияние на декодирование символов лишь в пределах одного блока. В случае же сверточных кодов, когда при кодировании каждый информационный символ воздействует на несколько блоков, возникновение одиночной ошибки при декодировании оказывает влияние на декодирование нескольких последующих блоков, а при некоторых неблагоприятных условиях и на бесконечное число блоков. В частности, в системах кодирования, использующих декодирование с обратной связью, при возникновении ошибки декодирования по неправильным оценкам значений ошибок $\hat{e}_0^{(j)}$, $j = 1, 2, \ldots, k_0$, осуществляется коррекция синдрома (естественно, неправильная), что может привести к новой ошибке декодирования. Таким образом, при построении сверточных кодов необходимо стремиться достичь не только высокой корректирующей способности, но и по возможности минимальной глубины распространения ошибок.

Глубина распространения ошибок при декодировании с обратной связью определяется структурой логических схем декодера, и поэтому анализ этой структуры обычно сводится к анализу функционирования регистров сдвига с нелинейной обратной связью, образующих декодирующие логические схемы. В случае, если шумы превышают корректирующую способность кода и это приводит к появлению ошибки декодирования, то обычно в течение некоторого периода времени ошибок не возникает, декодирующая логическая схема автоматически возвращается в нормальное рабочее состояние и распространение ошибок прекращается. В этом случае говорят о конечной глубине распространения ошибок. Если же декодирующая логическая схема в нормальное рабочее состояние не возвращается, то глубина распространения ошибок считается бесконечной. Таким образом, анализ распространения ошибок сводится к анализу работы регистров сдвига с нелинейной обратной связью, когда на их входы подаются нулевые символы, т. е. в режиме самоконтроля.

5.7.1. Анализ глубины распространения ошибок

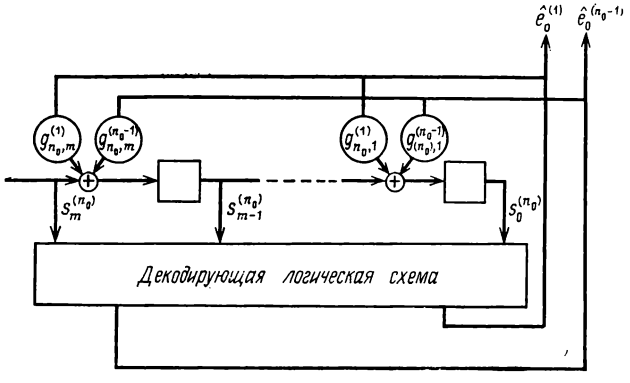
Для простоты рассмотрим случай, когда $k_0 = n_0 - 1$. Регистр сдвига, предназначенный для хранения символов синдрома, и логическая декодирующая схема декодера показаны на фиг. 5.5. В данном случае синдром

$$s^{(n_0)}(D) = s_0^{(n_0)} + s_1^{(n_0)} D + s_2^{(n_0)} D^2 + \ldots \qquad (5.59)$$

вычисляется в декодере согласно правилу

$$s^{(n_0)}(D) = \sum_{j=1}^{n_0-1} G^{(j)}_{(n_0)}(D)\, E^{(j)}(D) - E^{(n_0)}(D) \qquad (5.60)$$

и подается на вход регистра синдрома, показанного на фиг. 5.5. При исправлении случайных ошибок с помощью порогового декодирования логическая декодирующая схема в момент времени m по исправленному синдрому, символы которого получаются из символов синдрома $s_0^{(n_0)}$, $s_1^{(n_0)}$, ..., $s_m^{(n_0)}$ вычитанием информационных шумовых символов, возникающих до момента



Фиг. 5.5. Регистр синдрома и логическая часть декодирующей схемы.

0, формирует оценки $\hat{e}_0^{(1)}, \ldots, \hat{e}_0^{(n_0-1)}$ значений информационных шумовых символов в момент времени 0. Используемый при этом исправленный синдром

$$\delta(m) = [\delta_0(m), \delta_1(m), \ldots, \delta_m(m)] \qquad (5.61)$$

определяется следующими равенствами:

$$\delta_i(m) = s_i^{(n_0)} - \sum_{j=1}^{n_0-1} \sum_{u=i+1}^{m} g^{(j)}_{(n_0)} \hat{e}^{(j)}_{i-u} =$$

$$= \sum_{j=1}^{n_0-1} \sum_{u=0}^{m} g^{(j)}_{(n_0),\,u} e^{(j)}_{i-u} - e_i^{(n_0)} - \sum_{j=1}^{n_0-1} \sum_{u=i+1}^{m} g^{(j)}_{(n_0),\,u} \hat{e}^{(j)}_{i-u}. \qquad (5.62)$$

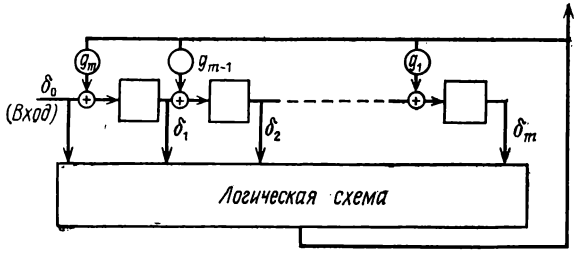
Если предположить, что предшествующие оценки шумовых символов были верными, то последнее равенство сведется к следующему:

$$\delta_i(m) = \sum_{j=1}^{n_0-1} \sum_{u=0}^{i} g^{(j)}_{(n_0),\,u} e^{(j)}_{i-u} - e_i^{(n_0)}. \qquad (5.63)$$

Величина $\delta_i(m)$, задаваемая этим равенством, равна символу синдрома

$$s_i^{(n_0)} = \sum_{j=1}^{n_0-1} \sum_{u=0}^{i} g_{(n_0),\,u}^{(j)} e_{i-u}^{(j)} - e_i^{(n_0)}. \qquad (5.64)$$

Если функцию F, реализуемую декодирующей логической схемой, выбрать таким образом, чтобы значение $F(s_0^{(n_0)}, \ldots, s_m^{(n_0)})$ давало правильные оценки символов $e_0^{(1)}, \ldots, e_0^{(n_0-1)}$, то при условии, что предыдущие оценки $\hat{e}_{-1}^{(j)}, \ldots, \hat{e}_{-m}^{(j)}$, $j = 1, \ldots, n_0 - 1$, правильные, эта функция будет давать правильные значения и $e_0^{(1)}, \ldots, e_0^{(n_0-1)}$. Однако если хотя бы одно из значений предшествующих оценок было правильное, то значения величин $\delta_i(m)$ будут отличаться от значений, даваемых формулой



Фиг. 5.6. Нелинейный регистр сдвига.

(5.63), и при декодировании с большой вероятностью оценки $\hat{e}_0^{(1)}, \ldots, \hat{e}_0^{(n_0-1)}$ будут неверными. Это в свою очередь порождает тенденцию к возникновению последовательности новых ошибок. В подобных случаях, когда конечное число ошибок в канале приводит к возникновению бесконечной последовательности ошибок декодирования, говорят, что декодирующая логическая схема с обратной связью имеет бесконечную глубину распространения ошибок.

Внутреннее состояние декодирующей логической схемы, показанной на фиг. 5.5, определяется состояниями $(m+1)$ ячеек регистра сдвига. Поэтому если после момента возникновения ошибки декодирования в течение m последующих моментов времени декодирование происходит правильно, то влияние этой ошибки декодирования исчезает и схема оказывается в нормальном рабочем состоянии. Следовательно, необходимым условием возникновения бесконечного распространения ошибки является возникновение после каждой ошибки декодирования в течение m последующих моментов времени новой ошибки декодирования.

Далее для простоты рассмотрим работу схемы, изображенной на фиг. 5.6, когда на вход регистра сдвига с нелинейной обратной связью подается последовательность из одних нулей, т. е. в режиме самоконтроля. Пусть $\boldsymbol{\delta} = (\delta_0, \delta_1, \ldots, \delta_m)$ — внутреннее состояние регистра сдвига и $\mathbf{0}$ — вектор, все компоненты которого равны нулю. Если обозначить через T оператор следующего состояния, то функционирование схемы, показанной на фиг. 5.6, можно описать равенством

$$T\boldsymbol{\delta} = \boldsymbol{F}(\boldsymbol{\delta}), \tag{5.65}$$

где $\boldsymbol{F}$ — некоторая векторная функция ($\boldsymbol{F}(\mathbf{0}) = \mathbf{0}$). Вектор $T\boldsymbol{\delta}$ представляет собой состояние нелинейного регистра сдвига после однократного сдвига, а $T^N\boldsymbol{\delta}$ — состояние после N последовательных сдвигов. Из того, что $\boldsymbol{F}(\mathbf{0}) = \mathbf{0}$, следует, что $T\mathbf{0} = \mathbf{0}$, т. е. $\mathbf{0}$ является равновесным состоянием регистра сдвига с нелинейной обратной связью.

Определение 5.3. *Если для каждого состояния* $\boldsymbol{\delta}$ *существует такое* N, *что* $T^N\boldsymbol{\delta} = \mathbf{0}$, *то регистр сдвига с нелинейной обратной связью называется устойчивым. В противном случае этот регистр называют неустойчивым.*

Для самоортогональных кодов существуют простые достаточные условия устойчивости, которые оказываются полезными и с практической точки зрения. Для произвольных кодов простых общих критериев устойчивости не существует. В настоящее время известны два метода определения устойчивости: метод Ляпунова, использованный Месси и Лиу [32], и метод, предложенный Робинсоном [3] для дефинитного декодирования.

5.7.2. Критерий устойчивости пороговой декодирующей логической схемы

Обозначим через $W(\boldsymbol{\delta})$ вес внутреннего состояния регистра сдвига с нелинейной обратной связью, который определим как число ненулевых компонент вектора $\boldsymbol{\delta} = (\delta_0, \delta_1, \ldots, \delta_m)$. Необходимое условие устойчивости пороговой декодирующей логической схемы состоит в следующем.

Критерий устойчивости. *Если каждый шаг декодирования, при котором значение оценки шумового символа равно 1, приводит к уменьшению веса внутреннего состояния, т. е. если при каждом таком шаге* $W(T\boldsymbol{\delta}) \leqslant W(\boldsymbol{\delta})$, *то пороговая декодирующая логическая схема является устойчивой.*

Доказательство. Распространение ошибки будет иметь место только в том случае, если после каждой ошибки декодирования при декодировании m последующих блоков возникнет хотя бы

одна новая ошибка декодирования. Однако в данном случае при этом вес внутреннего состояния обязательно будет уменьшаться. Максимальный вес внутреннего состояния равен m, так что если вес внутреннего состояния становится равным $[J/2]$ или меньшим, то значение оценки декодирования ни на каком из последующих шагов не может равняться 1. Следовательно, самое большее после $m(m+1-[J/2])$ сдвигов вес внутреннего состояния будет $[J/2]$ или меньше. Если вес внутреннего состояния не превышает $[J/2]$, то после самое большее $(m+1)$ сдвигов внутреннее состояние регистра сдвига с нелинейной обратной связью будет нулевым. Из вышеизложенного следует, что

$$N \leqslant m(m-[J/2]+2)+1. \tag{5.66}$$

Таким образом, мы доказали, что пороговая декодирующая логическая схема является устойчивой.

То, что самоортогональные коды имеют конечную глубину распространения ошибок, было показано Робинсоном и Бернстейном [3] непосредственно с помощью этого достаточного условия. Действительно, пусть V — минимальное число ненулевых значений входов порогового элемента, при котором выход порогового элемента равен 1. В случае самоортогональных кодов на входы порогового элемента подаются J символов синдромов. Если среди этих символов синдромов V или более равны 1, то выход декодирующей логической схемы оказывается равным 1 и используется для исправления ошибок и J символов синдромов. После исправления этих символов синдрома равными 1 будут не более чем $J-V$ из них. Поскольку в режиме самоконтроля на вход поступают только символы 0, то в результате таких операций ни один из этих вводимых по цепи обратной связи в синдром символов 1 не уничтожится, так что число символов 1 среди символов синдрома уменьшится по крайней мере на $V-(J-V)=2V-J$. Следовательно, необходимые условия устойчивости пороговой декодирующей логической схемы для самоортогональных кодов выполнены. Как показали Робинсон и Бернстейн, число L ошибочных символов, которые должны следовать за произвольной ошибкой декодирования для самоортогональных кодов, удовлетворяет следующему соотношению:

$$L \leqslant (n_A - n_0)t + 2n_A, \tag{5.67}$$

где t — максимальное целое число, удовлетворяющее неравенству

$$\binom{V+1}{2} + \binom{2V-J}{2} \geqslant (N-J+V-1)\left[\frac{J-V}{V}\right]^t.$$

Неравенство (5.66) гарантирует также устойчивость рассматриваемых в следующем разделе сверточных кодов, исправ-

ляющих пачки ошибок. Однако для ортогонализируемых кодов это достаточное условие может и не выполняться. Например, для рассматриваемого в разд. 5.3 кода со скоростью $R = 1/2$ и кодовым ограничением $n_A = 12$ при $e_0^{(1)} = e_1^{(1)} = 1$ символы синдрома $s_0^{(2)} = s_1^{(2)} = s_3^{(2)} = 1$ и вес внутреннего состояния равен 3. В этом случае $\hat{e}_0^{(1)} = 1$, но после сдвига внутренним состоянием регистра сдвига оказывается последовательность 1001100, имеющая также вес 3. В подобных случаях необходимо пользоваться более сложными критериями устойчивости. Одним из таких критериев является рассматриваемый ниже критерий, основанный на использовании функции Ляпунова.

5.7.3. Критерий, основанный на использовании функции Ляпунова [32]

Пусть c — произвольное множество векторов, а $\bar{c}$ — множество, являющееся дополнением c. Пусть $D(\boldsymbol{\delta})$ — функция, принимающая два значения и такая, что $D(\boldsymbol{\delta}) = 0$, если $\boldsymbol{\delta} \in c$, и $D(\boldsymbol{\delta}) = 1$, если $\boldsymbol{\delta} \in \bar{c}$. Изменение функции $D(\boldsymbol{\delta})$, связанное с изменением состояния $\boldsymbol{\delta}$ при сдвиге, обозначим через ΔD. Величина $\Delta D = 1$ тогда и только тогда, когда состояние переходит из c в $\bar{c}$ или, наоборот, из $\bar{c}$ в c. Следовательно, если состояние $\boldsymbol{\delta} = 0$ не изменяется, то $\Delta D = 0$. Аналогом теоремы Ляпунова об асимптотической устойчивости является следующая теорема.

Теорема 5.5. *Пусть* $D_0(\boldsymbol{\delta})$ *— функция, такая, что* $D_0(\mathbf{0}) = 0$ *и* $D_0(\boldsymbol{\delta}) = 1$ *при всех* $\boldsymbol{\delta} \neq 0$. *Тогда регистр сдвига с нелинейной обратной связью является неустойчивым, если* $\Delta D_0 = 0$ *при всех* $\boldsymbol{\delta}$, *и устойчивым, если* $\Delta D_0 = 1$ *при всех* $\boldsymbol{\delta} \neq 0$.

Эта теорема показывает, что регистр сдвига с нелинейной обратной связью является устойчивым, если при сдвиге любое его ненулевое внутреннее состояние переходит в состояние **0**, и неустойчивым, если при сдвиге никакое состояние, кроме, быть может, нулевого, не переходит в состояние **0**. Однако, как правило, в действительности часть ненулевых состояний при сдвиге переходит в состояние **0**, но существуют также так называемые неопределенные состояния, которые при сдвиге в состояние **0** не переходят. Для определения устойчивости при наличии таких неопределенных состояний можно воспользоваться следующим алгоритмом:

1) Пусть c_0 — множество, включающее только состояние **0**.

2) Определим функцию $D_0(\boldsymbol{\delta})$ так, как указано в теореме 5.5.

3) Пусть s_1 — множество всех таких $\boldsymbol{\delta}$, что $\Delta D_0(\boldsymbol{\delta}) = 1$, и пусть $c_1 = c_0 \cup s_1$. Состояния, принадлежащие s_1, обладают тем свойством, что при сдвиге они переходят в состояние **0**. Если $\bar{c}_1$ яв-

ляется пустым множеством, т. е. если $\bar{c}_1 = \phi$, то из теоремы 5.5 следует, что рассматриваемая схема устойчивая, а если $s_1 = \phi$, то эта схема неустойчивая. Если ни одно из этих условий не выполняется, то переходим к следующему шагу:

4) Определим функцию $D_1(\delta)$: $D_1(\delta) = 0$, если $\delta_1 \in c_1$, и $D_1(\delta) = 1$, если $\delta \in \bar{c}_1$.

5) Пусть s_2 — множество всех состояний δ, для которых $\Delta D_1(\delta) = 1$, и пусть $c_2 = c_1 \cup s_2$. К s_2 относятся те состояния, которые переходят в состояние **0** в результате двух сдвигов.

6) Этот процесс будем продолжать до тех пор, пока одно из множеств $\bar{c}_M$ или s_M не окажется пустым (заметим, что для этого потребуется конечное число шагов M, а именно $M \leqslant 2^m$). Если $\bar{c}_M = \phi$, т. е. если все состояния после M или менее сдвигов переходят в состояние **0**, то схема является устойчивой. Если $s_M = \phi$, то существует некоторое ненулевое состояние, которое никогда в состояние **0** не переходит, что означает неустойчивость исследуемой схемы.

Используемые в этом алгоритме функции D_i можно определить также следующим образом. Пусть

$$D_0 = 1 \oplus \prod_i (\delta_i \oplus 1) \tag{5.68}$$

и

$$P_j(\delta) = \sum_{\delta \in s_j} \prod_i (\delta_i \oplus \delta_i' \oplus 1). \tag{5.69}$$

Тогда

$$D_n(\delta) = D_0(\delta) \oplus \sum_{j=1}^{n} P_j, \quad n = 1, 2, \ldots, M, \tag{5.70}$$

и

$$\Delta D_n(\delta) = P_n(T\delta). \tag{5.71}$$

Эти формулы позволяют упростить вычисления при определении устойчивости той или иной схемы с помощью описанного выше алгоритма.

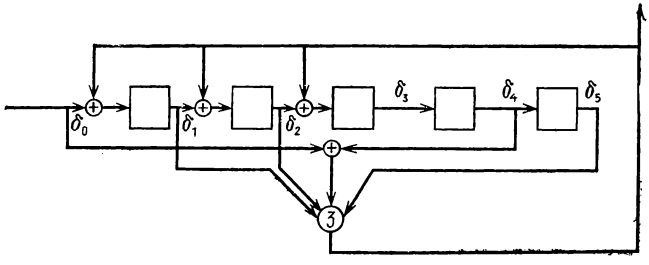
Пример 5.8. Воспользуемся описанным выше алгоритмом для определения устойчивости декодирующей логической схемы ортогонализуемого кода из примера 5.6. Регистр сдвига с нелинейной обратной связью в этом случае имеет вид, показанный на фиг. 5.7. В целях упрощения внутренние состояния этого регистра будем представлять в десятичной записи, а именно состоянию $\delta = (\delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5)$ сопоставим десятичное число $\delta_1 + \delta_2 2 + \delta_3 2^2 + \delta_4 2^3 + \delta_5 2^4$ (δ_0 в режиме самоконтроля всегда равно 0). Состояния ячеек регистра, которые могут принимать значения и 0 и 1, будем обозначать через $\times$. Основываясь на фиг. 5.7, можно установить следующее:

1) При однократном сдвиге только состояние $\delta = (0, 0, 0, 0, 1)$, десятичным представлением которого является число 16, переходит в состояние **0**.

2) При однократном сдвиге в состояние 16 переходят лишь состояния вида $\delta = (0, 0, 0, 1, \times)$, т. е. состояния 8 и 24.

3) При однократном сдвиге в состояние 8 или 24 переходят лишь состояния вида $\delta = (0, 0, 1, \times, \times)$, т. е. состояния 4, 20, 12 и 28.

4) При однократном сдвиге в состояния 4, 20, 12 и 28 переходят лишь состояния вида $\delta = (0, 1, \times, 0, 1)$, $\delta = (0, 1, \times, 1, 0)$ и $\delta = (0, 1, \times, 0, 0)$, т. е. состояния 2, 18, 10, 6, 22, 24.



Фиг. 5.7. Регистр синдрома (12,6)-кода.

Продолжая этот процесс дальше, построим табл. 5.4 и найдем, что $s_9 = \phi$. Это означает, что рассматриваемая схема является неустойчивой.

Однако вывод о неустойчивости схемы, изображенной на фиг. 5.7, не означает, что эта схема имеет бесконечную глубину распространения ошибок. Декодирующая логическая схема всегда начинает работать с нулевого внутреннего состояния и в зависимости от значений символов синдрома, поступающих на ее вход, проходит те или иные внутренние состояния. Следовательно, из всех внутренних состояний регистра сдвига с нелинейной обратной связью с распространением ошибок связаны лишь так называемые достижимые состояния, т. е. состояния, в которых регистр может оказаться при всевозможных синдромах на его входе. Здесь вместо определения 5.3, которое понятие устойчивости связывало со всеми внутренними состояниями, рассмотрим следующее определение.

Определение 5.4. (Месси, Лиу.) *Регистр сдвига с нелинейной обратной связью называется устойчивым по отношению к входному воздействию тогда и только тогда, когда для любого состояния* δ, *которое этот регистр может достичь из состояния* 0

Таблица 5.4

Определение устойчивости схемы, изображенной на фиг. 5.7

j	s_j	Число элементов $\bar{c}_j$	$p_j(s)$
1	16	30	$\delta_5(\delta_4+1)(\delta_3+1)(\delta_2+1)(\delta_1+1)$
2	8, 24	28	$\delta_4(\delta_3+1)(\delta_2+1)(\delta_1+1)$
3	4, 20, 12, 28	24	$\delta_3(\delta_2+1)(\delta_1+1)$
4	2, 18, 10, 6, 22, 14	18	$(\delta_4\delta_5+1)\delta_2(\delta_1+1)$
5	1, 17, 9, 5, 21, 3, 7	11	$\delta_1[(\delta_2+1)(\delta_3+1)\delta_4(\delta_5+1)+$ $+\delta_2(\delta_4+1)(\delta_5+1)+(\delta_2+1)(\delta_4+1)]$
6	25, 17, 11, 27, 23	6	$\delta_1[\delta_2(\delta_3+1)\delta_4+\delta_2+\delta_3(\delta_4+1)\delta_5+$ $+\delta_2\delta_5(\delta_3+1)(\delta_3+1)+\delta_4\delta_5(\delta_3+1)(\delta_2+1)]$
7	26, 30, 15, 31	2	$\delta_4\delta_2[\delta_5(\delta_1+1)+\delta_3\delta_1]$
8	13	1	$\delta_1(\delta_2+1)\delta_4\delta_3(\delta_5)$
9	ϕ	1	Вывод: неустойчива

при той или иной входной последовательности, существует такое число N, что $T^N\boldsymbol{\delta}=\mathbf{0}$.

Если регистр сдвига с нелинейной обратной связью является устойчивым по отношению к входному воздействию, то глубина распространения ошибок бесконечной быть не может. Наоборот, если регистр сдвига с нелинейной обратной связью не является устойчивым по отношению к входному воздействию, то глубина распространения ошибок может оказаться и бесконечной. Заметим, что если регистр сдвига с нелинейной обратной связью является просто устойчивым, то он устойчив и по отношению к входному воздействию. Обратное утверждение неверно. Действительно, как было показано выше, регистр сдвига, изображенный на фиг. 5.7, не является устойчивым, поскольку из состояния 29 нельзя достичь состояния **0**. Из состояния 29 этот регистр вновь переходит в состояние 29, т. е. это состояние является равновесным состоянием. При определенных последовательностях на входе в состояние 29 можно попасть из состояний 30 и 14. В состояние 14 можно попасть только из состояния 7, но состояние 7 является недостижимым (т. е. в него нельзя попасть из состояния **0**). Кроме того, недостижимым является и состояние 30. Следовательно, регистр сдвига с нелинейной обратной связью, изображенный на фиг. 5.7, является устойчивым по отношению к входному воздействию и имеет конечную глубину распространения ошибок.

5.7.4. Дефинитное декодирование

Другим методом борьбы с распространением ошибок является так называемый метод дефинитного декодирования. При этом декодировании обратного воздействия на синдром нет (т. е. исправление синдрома не проводится) и оценка значения информационного символа в момент u осуществляется по информационным символам, принятым с момента времени $u-m$ до момента времени $u+m$, и проверочным символам, принятым с момента времени u до момента времени $u+m$.

Поскольку кодовое ограничение такого кода n_{DD} равно $mk_0+(m+1)n_0$ и обратная связь отсутствует, то влияние возникшей ошибки декодирования распространяется не более чем на декодирование n_{DD} символов. Однако при таком декодировании и достаточно больших кодовых ограничениях корректирующие способности кодов обычно оказываются хуже, чем при декодировании с обратной связью.

Различные свойства дефинитного декодирования методами теории матриц установил Робинсон [33]. По существу эти же методы теории матриц использовали Вайнер и Эш [6] для анализа корректирующих способностей сверточных кодов при декодировании с обратной связью. Этот анализ интересен по многим причинам, но так как коды специально для дефинитного декодирования почти не разрабатывались (ввиду их несколько худших корректирующих способностей), то здесь эти вопросы подробно не рассматриваются.

Как было показано выше, для сверточных кодов характерно распространение ошибок. Поэтому всегда хотелось бы использовать коды, для которых глубина распространения ошибок небольшая. Следует, однако, заметить, что реально возникновение ошибок в количестве, превышающем корректирующие способности кода, не всегда приводит к распространению ошибок; возникнет или не возникнет распространение ошибок, зависит от того, какова конфигурация ошибок, появившихся в канале.

5.8. Сверточные коды, исправляющие пачки ошибок

Несколько раньше, чем было введено пороговое декодирование, Хегельбергер [4] построил сверточные коды, исправляющие пачки ошибок. В дальнейшем Вайнер и Эш [6], Препарата [7], Берлекэмп [5] и Месси [8] получили нижние границы для длины свободного от ошибок защитного интервала, необходимого для исправления ошибок, разработали методы кодирования, основанные на теории матриц, и методы декодирования, использующие структуру систематических кодов.

В данном разделе будут рассмотрены так называемые коды Ивадари, исправляющие пачки ошибок и асимптотически достигающие нижней границы для кодового ограничения (или для длины защитного интервала) кодов, исправляющих пачки ошибок [10—12]. Эти коды могут быть очень просто реализованы, поскольку метод их декодирования является самым простым, который только может быть при пороговом декодировании.

При построении кодов, исправляющих пачки ошибок, как и в случае блоковых кодов, очень важно иметь нижнюю границу для кодового ограничения n_A (или длины защитного интервала). Защитным интервалом называют последовательность свободных от ошибок символов, следующих за пачкой ошибок. Защитный интервал необходим для того, чтобы пачки ошибок можно было исправлять. Вайнер и Эш [6] получили следующую границу для кодового ограничения (и длины защитного интервала g) кода, исправляющего пачки ошибок длины b:

$$n_A \geqslant g + b \geqslant 2b/(1 - R). \tag{5.72}$$

Рассматриваемые в этом разделе коды Ивадари представляют собой подкласс сверточных кодов, в котором для любых целых чисел $n_0 \geqslant 2$ и $\beta \geqslant 1$ существует код со скоростью $R = (n_0 - 1)/n_0$, исправляющий любые пачки ошибок длины $b = n_0\beta$ и менее. Любой код из этого класса, исправляющий пачки ошибок длины $b = n_0$, называется основным кодом. Коды со скоростью $R = (n_0 - 1)/n_0$, исправляющие пачки ошибок длины $b = \beta n_0$, $b > 1$, получаются из основного кода методом чередования.

5.8.1. Метод чередования

Существуют два типа метода чередования: простой и обобщенный. При простом чередовании с периодом x кода с порождающими многочленами $G_{(i)}^{(j)}(D)$, $j = 1, \ldots, k_0$, $i = 1, \ldots, n_0$, все показатели степеней в каждом порождающем многочлене умножаются на число x. В результате простого чередования кода с проверочной матрицей H получается новый код, проверочная матрица которого может быть получена из матрицы H путем введения между любыми двумя соседними строками последней $x - 1$ нулевых строк. Если же на x умножаются не все показатели степеней в порождающих многочленах, а только те, которые принадлежат некоторому заданному множеству, то такое чередование называется *обобщенным*. При обобщенном чередовании проверочная матрица получающегося кода строится из проверочной матрицы исходного кода путем введения $x - 1$ нулевых строк между каждой строкой из некоторого заданного подмножества строк и следующей строкой.

Пример 5.9. Рассмотрим сверточный код из примера 5.6 со скоростью $R = 1/2$ и кодовым ограничением $n_A = 12$, который задается порождающим многочленом

$$G_{(2)}^{(1)}(D) = 1 + D^3 + D^4 + D^5. \tag{5.73}$$

Матрица H для этого кода имеет следующий вид:

$$H = \begin{bmatrix} 1 & 1 & & & & & & & & & & \\ 0 & 0 & 1 & 1 & & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & 1 & & & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

В результате простого чередования с периодом 2 этого кода получается код с порождающим многочленом

$$G_{(2)}^{(1)}(D) = 1 + D^6 + D^8 + D^{10}. \tag{5.74}$$

Этот код имеет следующую матрицу:

$$H = \begin{bmatrix} 1 & 1 & & & & & & & & & & & & & & & & & & & & \\ 0 & 0 & 1 & 1 & & & & & & & & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & & & \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

При обобщенном чередовании исходного кода, а именно при чередовании членов с нулевой, второй и третьей степенью, получается код с порождающим многочленом

$$G_{(2)}^{(1)}(D) = 1 + 0 \cdot D + 0 \cdot D^2 + 0 \cdot D^3 + 0 \cdot D^4 + 0 \cdot D^5 + \\ + 0 \cdot D^6 + D^7 + D^8 = 1 + D^5 + D^7 + D^8.$$

Матрица H получающегося кода имеет следующий вид:

$$H = \begin{vmatrix}
1 & 1 & & & & & & & & & & & & & & & & & & \\
0 & 0 & 1 & 1 & & & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & & & & & & \mathbf{0} & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & & & \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & & & \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & & & \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & & & & \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & &
\end{vmatrix}.$$

Как видно из этого примера, при обобщенном чередовании нужно всегда указывать, какие слагаемые порождающего многочлена или какие строки матрицы участвуют в чередовании.

5.8.2. Принцип построения кодов

Основная идея, использованная Ивадари, заключалась в том, чтобы построить коды, удовлетворяющие следующим требованиям. При отсутствии ошибок в канале синдром должен быть нулевым. Любая ошибка, возникшая в любой передаваемой последовательности, должна иметь в синдроме так называемую собственную часть, не пересекающуюся с собственными частями других ошибок. Кроме того, код должен быть построен таким образом, чтобы декодер мог выделить собственную часть любой ошибки, а следовательно, обнаружить и исправить эти ошибки.

При построении кодов Ивадари $n_0 - 1$ информационным последовательностям сопоставляются различные последовательности веса 2 вида (e, e), $(e, 0, e)$, $(e, 0, 0, e)$, ..., где e — элемент поля $GF(q)$; единственной проверочной последовательности при этом сопоставляется последовательность $(e, 0, \ldots, 0)$ веса 1. Последовательности $(e, 0, \ldots, 0, e)$ длины k называются единичными конфигурациями длины k, а последовательности $(e, 0, \ldots, 0)$ длины k — проверочными конфигурациями длины k. Если информационным последовательностям сопоставить двоичные представления (011), (101), ... нечетных целых чисел, а проверочной последовательности — проверочную конфигурацию, то точно так же, как и коды Ивадари, можно построить коды Хегельбергера. Метод декодирования всех этих кодов один и тот же.

Проще всего коды Ивадари можно ввести, если описать структуру синдрома этих кодов. Поскольку при возникновении пачки ошибок длины βn_0 в каждой из передаваемых последова-

тельностей может оказаться β ошибок, то при возникновении такой пачки ошибок β собственных конфигураций ошибок для каждой из n_0 передаваемых последовательностей должны появиться в синдроме. Задача построения кода, удовлетворяющего указанным выше требованиям, сводится к тому, чтобы разместить эти конфигурации в синдроме так, чтобы они не накладывались друг на друга и при декодировании их можно было бы безошибочно выделить. В целях упрощения последовательность ошибок, возникающих в j-й передаваемой последовательности с момента времени 0 до момента времени $\beta - 1$, будем обозначать в дальнейшем как

$$e_0^{(j)} e_1^{(j)} \ldots e_{\beta-1}^{(j)} = E^{(j)}, \tag{5.75}$$

где $e_i^{(j)}$ — ошибка в момент времени i в j-й передаваемой последовательности.

5.8.3. Способ построения кодов

Рассматриваемые в этом разделе коды большей частью строятся методом проб и ошибок; общим достоинством этих кодов является то, что их структура позволяет сначала исправлять ошибки, возникшие в $(n_0 - 1)$-й информационной последовательности, затем ошибки, появившиеся в $(n_0 - 2)$-й информационной последовательности, и наконец ошибки, возникшие в 1-й информационной последовательности. Это позволяет правильно исправить ошибки даже в случае, если они возникли за пределами 1-й информационной последовательности. При этом используется свойство ошибок возникать пачками. Следует также обратить внимание на то, что если при декодировании рассматриваемых здесь кодов ошибки различных информационных последовательностей исправляются раздельно, то при декодировании рассмотренных ранее самоортогональных и ортогонализируемых кодов, предназначенных для исправления случайных ошибок, сначала исправляются ошибки, возникшие в момент времени 0 во всех $n_0 - 1$ информационных последовательностях, затем одновременно исправляются ошибки, возникшие в $n_0 - 1$ информационных последовательностях в момент времени 1, и т. д.

Сначала рассмотрим коды типа I, на основе которых строятся коды типа II. Сопоставив n_0-й передаваемой последовательности проверочную конфигурацию длины $\beta + 1$, а $(n_0 - j)$-й передаваемой последовательности единичную конфигурацию длины $\beta + j$, $j = 1, 2, \ldots, n_0 - 1$, сформируем

синдром следующим образом:

$$s_{\mathrm{I}}^{(n_0)}(D) = E^{(n_0)} \underbrace{0\,0\,\ldots\,0}_{\beta}\ \underbrace{0\,0\,\ldots\,0}_{x}\ E^{(n_0-1)}E^{(n_0-1)}E^{(n_0-2)}\,0\,E^{(n_0-2)}\ldots$$

$$\ldots E^{(1)}\ \underbrace{0\,0\,\ldots\,0}_{n_0-2}\ E^{(1)}. \qquad (5.76)$$

Заметим, что в эту формулу входят все $n\beta$ конфигураций ошибок. Предполагая, что используется метод декодирования с обратной связью, выберем параметр x таким образом, чтобы в синдроме можно было при декодировании выделить без ошибок собственные конфигурации каждой передаваемой последовательности.

Чтобы разложить при декодировании ненулевой синдром на $n_0 - 1$ различных единичных конфигураций, используется следующий алгоритм декодирования. С появлением каждого ненулевого символа в синдроме начинается поиск единичных конфигураций, образованных этим и предшествующими ненулевыми символами, хранящимися в ячейках памяти декодирующей логической схемы.

1) Если ни одну из $n_0 - 1$ единичных конфигураций обнаружить не удается, то этот ненулевой символ синдрома сдвигается в регистр.

2) Если обнаружена одна из $n_0 - 1$ единичных конфигураций, по длине этой конфигурации определяется номер переданной последовательности, в которой возникла ошибка, породившая эту конфигурацию, а символ, входящий в эту конфигурацию, дает значение ошибки. Это позволяет исправить обнаруженную ошибку и провести исправление синдрома.

3) Если одновременно можно построить несколько единичных конфигураций, то с помощью указанного ниже правила запрета выбирается одна из этих конфигураций и осуществляется переход к шагу 2.

Правило запрета: «Если обнаружено несколько единичных конфигураций, то выбирается та, которая имеет бóльшую длину».

Последовательность ошибок $E^{(n_0)}$, возникших в n_0-й передаваемой последовательности (проверочной последовательности), для исправления ошибок в информационных передаваемых последовательностях не используется. Величину x следует выбрать так, чтобы указанные выше операции 1—3 выполнялись без ошибок и длина синдрома была бы по возможности минимальной. Этим условиям удовлетворяет $x = n_0 - 2$. Таким образом строятся коды типа I.

5.8.4. Конкретные коды

Код типа I [10—12]:

$$s_{\mathrm{I}}^{(n_0)}(D) = E^{(n_0)} \underbrace{0\,0\,\ldots\,0}_{\beta + n_0 - 2} E^{(n_0-1)}E^{(n_0-1)}\,0\,E^{(n_0-2)}\,0\,E^{(n_0-2)}\,\ldots$$
$$\ldots\,E^{(1)} \underbrace{0\,0\,\ldots\,0}_{n_0-2} E^{(1)}. \qquad (5.77)$$

Порождающими многочленами этого кода являются многочлены

$$G_{(n_0)}^{(j)}(D) = D^{n_0-2+2\beta(n_0-j)+\sum_{i=1}^{n_0-2-j} i}\left(1 + D^{\beta+n_0-j-1}\right). \qquad (5.78)$$

Используя тот же самый метод декодирования, что и для кода типа I, но несколько изменяя расположение в синдроме последовательностей $E^{(j)}$, можно получить следующий код.

Код типа II [11, 12]:

$$s_{\mathrm{II}}^{(n_0)}(D) = E^{(n_0)}E^{(n_0-1)}\,0\,E^{(n_0-2)}\,0\,\ldots$$
$$\ldots\,0\,E^{(1)} \underbrace{0\,0\,\ldots\,0}_{n_0+\beta-1} E^{(n_0-1)}E^{(n_0-2)}\,\ldots\,E^{(1)}. \qquad (5.79)$$

Порождающими многочленами кода типа II являются многочлены

$$G_{(n_0)}^{(j)}(D) = D^{\beta(n_0-j)+n_0-j-1}\left(1 + D^{n_0(\beta+1)+j-2}\right). \qquad (5.80)$$

Принцип декодирования кодов типа I и типа II один и тот же, но тем не менее эти коды несколько отличаются друг от друга.

Пример 5.10 иллюстрирует процесс декодирования рассмотренных выше кодов.

Пример 5.10. Рассмотрим код Ивадари типа II с параметрами $n_0 = 3$, $\beta = 2$, $q = 2$. Согласно формуле (5.80), этот код задается порождающими многочленами

$$G_{(3)}^{(1)}(D) = D^5\left(1 + D^8\right) = D^5 + D^{13}, \qquad (5.81)$$

$$G_{(3)}^{(2)}(D) = D^2\left(1 + D^9\right) = D^2 + D^{11}. \qquad (5.82)$$

Предположим, что возникли следующие ошибки:

$$\left.\begin{aligned} E^{(1)}(D) &= e_1^{(1)}D + e_2^{(1)}D^2, \\ E^{(2)}(D) &= e_0^{(2)} + e_1^{(2)}D, \\ E^{(3)}(D) &= e_0^{(3)} + e_1^{(3)}D. \end{aligned}\right\} \qquad (5.83)$$

Конечно, на приемном конце эти многочлены неизвестны, но может быть вычислен синдром, который, согласно формулам (5.81)—(5.83), в данном случае имеет следующий вид:

$$s_{II}^{(3)}(D) = \sum_{j=1}^{2} E^{(j)}(D)\, G^{(j)}(D) - E^{(3)}(D) =$$
$$= -e_0^{(3)} - e_1^{(3)} D_0 e^{(2)} D^2 + e_1^{(2)} D^3 + e_1^{(1)} D^6 + e_2^{(1)} D^7 +$$
$$+ e_0^{(2)} D^{11} + e_1^{(2)} D^{12} + e_1^{(1)} D^{14} + e_2^{(1)} D^{15} =$$
$$= E^{(3)} E^{(2)} 00 E^{(1)} 000 E^{(2)} 0 E^{(1)} =$$
$$= e_0^{(3)} e_1^{(3)} e_0^{(2)} e_1^{(2)} 00 e_1^{(1)} e_2^{(1)} 000 e_0^{(2)} e_1^{(2)} 0 e_1^{(1)} e_2^{(1)}. \qquad (5.84)$$

Из формул (5.81) и (5.82) следует, что единичные конфигурации второй и первой информационных последовательностей имеют длину соответственно 10 и 9.

Согласно шагу 1 алгоритма декодирования, до появления 2-го символа $e_0^{(2)}$ его единичная конфигурация образована быть не может и до этого момента поступающие символы ошибок просто направляются в регистр сдвига. При появлении 2-го символа $e_0^{(2)}$ алгоритм обнаружит две единичные конфигурации длины 10 и 9, образованные этим символом соответственно с 1-м символом $e_0^{(2)}$ и с 1-м символом $e_1^{(2)}$. Согласно правилу запрета, алгоритм выбирает конфигурацию длины 10, исправляет $e_0^{(2)}$ и устраняет воздействие $e_0^{(2)}$ на $s_{II}^{(3)}(D)$. После этого синдром принимает следующий вид:

$$s_{II}^{(3)}(D) = -e_0^{(3)} - e_1^{(3)}(D) + e_0^{(2)} D^2 (1 + D^9) +$$
$$+ e_1^{(2)} D^3 (1 + D^9) + e_1^{(1)} D^6 (1 + D^8) + e_2^{(1)} D^7 (1 + D^8).$$

Разлагая этот синдром дальше, можно исправить оставшиеся ошибки.

5.8.5. Оценки кодов и сравнение их параметров

Обычно используются следующие критерии оценки качества кодов, исправляющих пачки ошибок:

1) Достигает (быть может, асимптотически) или нет кодовое ограничение (или длина защитного интервала) нижней границы (5.72). Свойства сверточных кодов таковы, что соотношения (5.72) для этих величин со знаком равенства выполняться не могут. Так, Вайнер и Эш [3], рассматривая случай, когда пачка ошибок начинается за первой информационной последовательностью, получили следующую более точную границу для сверточных кодов:

$$n_A \geqslant g + b \geqslant \frac{2b}{1-R} + n_0 - 1 = 2bn_0 + n_0 - 1 \qquad (5.85)$$

и назвали оптимальными коды, для которых эти соотношения выполняются со знаком равенства. Следуя этому определению, будем называть квазиоптимальными коды, для которых

$$n_A = 2bn_0 + \text{const},$$

$$\text{const} > n_0 - 1,$$

т. е. нижняя граница (5.85) достигается асимптотически при большой длине исправляемых пачек ошибок.

2) Экономичность можно характеризовать числом каскадов регистров сдвига, используемых в кодере и декодере.

3) В случае возникновения ошибки декодирования глубина распространения последней должна быть по возможности меньшей. При декодировании рассматриваемых здесь кодов решение о наличии или отсутствии ошибки в символе принимается элементом И. Поскольку этот элемент можно рассматривать как пороговый элемент с двумя входами и порогом 1,5, то в данном случае можно воспользоваться критерием устойчивости пороговой декодирующей логической схемы. Так как при каждом срабатывании декодирующей логической схемы вес внутреннего состояния этой схемы уменьшается на 1 или 2, то, следовательно, схема устойчива. Что касается глубины распространения ошибок, то обычный защитный интервал длины g гарантирует, что влияние возникшей ошибки распространяется только на последующие ошибки в синдроме, а дополнительный защитный интервал $g_{\text{доп}}$ гарантирует, что все ненулевые символы будут из декодирующей логической схемы удалены. Сле-

Таблица 5.5

Необходимое число ячеек регистров сдвига и длина защитных интервалов кодов, исправляющих пачки ошибок

Код / Длина пачки ошибок	Защитный интервал		Необходимое число ячеек регистров сдвига	
	I	II	I	II
3	23	26	23	30
30	158	161	167	192
300	1 508	1 511	1 607	1 812
5	94	84	94	92
50	499	489	508	542
500	4 549	4 539	4 648	5 042
10	639	369	639	387
100	2 349	2 079	2 358	2 187
1000	19 449	19 179	19 548	20 187

довательно, глубина распространения ошибок не превышает $g + g_{\text{доп}}$.

В табл. 5.5 приведены значения указанных выше характеристик для кодов типа I и II.

5.8.6. Другие коды

Выше был описан метод построения кодов со скоростью $R = (n_0 - 1)/n_0$, $n_0 \geqslant 2$. Известно, что при $R = 1/n_0$ эти коды оказываются почти тривиальными, допускающими мажоритарное декодирование [10, 12].

Рассмотренные коды позволяют исправлять любые пачки ошибок длины b и менее. Однако известны коды [41], которые позволяют исправлять почти все пачки ошибок длины до $b = \beta n_0$, имеют скорость $R = (n_0 - 1)/n_0$ и защитный интервал, несколько больший чем $n_0(n_0 - 1)\beta$. Эти коды имеют несколько лучшие корректирующие способности, благодаря чему необходим меньший защитный интервал. Кроме того, эти коды позволяют исправлять также и случайные ошибки.

Найти регулярный метод построения циклических кодов с числом проверочных символов $m = 2b + \sigma$, $\sigma \ll 2b$, довольно трудно. Как указывается в гл. 7 и как в этом действительно нетрудно убедиться, рассмотренные коды соответствуют блоковым кодам с числом проверочных символов $m = 2b + \sigma$, $\sigma \ll 2b$. То, что эти коды можно строить регулярным методом, объясняется тем, что они имеют очень простые алгоритмы кодирования и декодирования и, кроме того, позволяют изменять избыточность. Коды с числом избыточных символов $2b$ являются наилучшими среди кодов, имеющих ту же длину и исправляющих пачки ошибок длины b. Известно, что если число проверочных символов больше $2b$, то вероятность возникновения в пределах одного блока или кодового ограничения двух пачек ошибок становится большой, а следовательно, становится большой и вероятность ошибочного декодирования.

5.9. Сверточные коды, исправляющие пачки ошибок и независимые ошибки (диффузные коды)

Рассматриваемые в этом разделе коды позволяют с помощью порогового декодирования исправлять как пачки ошибок, так и независимые ошибки. Эти коды требуют для исправления пачек ошибок несколько больший защитный интервал, чем коды, исправляющие только пачки ошибок, и в этом смысле они несколько хуже последних, но могут исправлять любые две независимые ошибки. Коленбург [9] и Месси пер-

выми исследовали такие коды и назвали их диффузными. Вначале эти коды были построены лишь для скорости $R = 1/2$ и $J = 4$, однако в дальнейшем Тонг [13] предложил относительно регулярный метод их построения и при других значениях параметров R и J.

5.9.1. Два типа пачек ошибок

Диффузные коды в отличие от кодов, рассматривавшихся в предыдущем разделе, все ошибки, возникшие в блоке, исправляют одновременно точно так же, как самоортогональные коды исправляют независимые ошибки. При изучении таких кодов удобно пользоваться следующим определением пачки ошибок.

Конфигурация 1, $b = 5$	× × × × ×	0 0 0 0 0	.....
Конфигурация 2, $b = 2$	0 0 × × 0	0 0 0 0 0	.....
Конфигурация 3, $b = 2$	0 0 0 0 ×	× 0 0 0 0	.....

Фиг. 5.8. Пачки ошибок типа B1 и B2.

Определение 5.5. *Пачкой ошибок типа* B2 *длиной* B *называется совокупность ошибок, расположенных в* B *следующих подряд блоках, первый и последний из которых содержат соответственно первый и последний ошибочные символы.*

Чтобы отличать пачки ошибок типа B2 от обычных пачек ошибок, последние называют *пачками ошибок типа* B1. На фиг. 5.8 показано несколько примеров пачек ошибок в случае $n_0 = 5$. Здесь через × обозначены ошибочные символы, а через 0 — символы, свободные от ошибок. Конфигурация 1 на фиг. 5.8 представляет собой пачку ошибок типа B1 длиной 5 бит, но в то же время является пачкой ошибок типа B2 длиной 1 блок. Конфигурация 2 представляет собой пачку ошибок типа B1 длиной 2 бита и пачку ошибок типа B2 длиной 1 блок. И наконец, конфигурация 3, как и конфигурация 2, является пачкой ошибок типа B1 длиной 2 бита, но как пачка ошибок типа B2 она имеет длину 2 блока. Другими словами, пачка ошибок длиной b бит является пачкой ошибок типа B2 длиной $]b/n_0[$ или $]b/n_0[+ 1$ блоков. И, наоборот, любая пачка ошибок типа B2 длиной B блоков является пачкой ошибок типа B1 длиной b бит, где $Bn_0 \leqslant b \leqslant (B+1)n_0$.

5.9.2. Корректирующая способность

Корректирующая способность диффузных кодов определяется следующей теоремой.

Теорема 5.6. *Сверточный код исправляет конфигурацию ошибок E, если шумовые символы* $e_0^{(j)}$, $j = 1, 2, \ldots, n_0 - 1$, *воздействующие на информационные символы нулевого блока, удовлетворяют следующим условиям:*

1) *при* $e_0^{(j)} = 0$ *воздействию E подвержены не более чем* $J/2$ *проверочных соотношений, в которые входит* $e_0^{(j)}$;

2) *при* $e_0^{(j)} = 1$ *не более чем* $(J/2) - 1$ *проверочных соотношений, в которые входит* $e_0^{(j)}$, *подвержены воздействию шумовых символов, входящих в E и отличных от* $e_0^{(j)}$.

Доказательство.

1) В этом случае число составных проверок A_j, $j = 1, 2, \ldots, J$, принимающих отличные от нуля значения, самое большее равно $J/2$ и не достигает порога $(J/2) + 0{,}5$.

2) Число проверок A_j, принимающих ненулевые значения, здесь не меньше чем $J - (J/2) - 1 = (J/2) + 1$, и, следовательно, ошибка исправляется.

5.9.3. Простой пример

Примером диффузного кода со скоростью $R = 1/2$, исправляющего любые две независимые ошибки или пачки ошибок длиной 2 при наличии защитного интервала длины $6b - 2$, является код, задаваемый порождающим многочленом

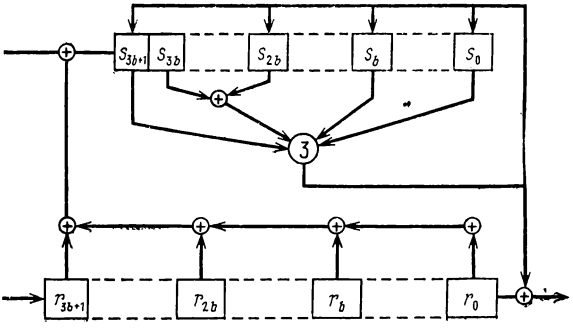
$$G_{(2)}^{(1)}(D) = 1 + D^b + D^{2b} + D^{3b+1}. \tag{5.86}$$

Декодер этого кода, показанный на фиг. 5.9, формирует четыре составные проверки:

$$\begin{aligned} A_1 &= s_0^{(2)} = e_0^{(1)} + e_0^{(2)}, \\ A_2 &= s_b^{(2)} = e_0^{(1)} + e_b^{(1)} + e_b^{(2)}, \\ A_3 &= s_{2b}^{(2)} + s_{3b}^{(2)} = e_0^{(1)} + e_{2b}^{(2)} + e_{3b}^{(1)} + e_{3b}^{(2)}, \\ A_4 &= s_{3b+1}^{(2)} = e_0^{(1)} + e_{b+1}^{(1)} + e_{2b+1}^{(1)} + e_{3b+1}^{(1)} + e_{3b+1}^{(2)}, \end{aligned} \tag{5.87}$$

ортогональные относительно $e_0^{(1)}$. Это означает, что с помощью порогового декодирования такой код позволяет исправить любые две или менее случайные ошибки. Предположим, что появилась пачка ошибок длиной $2b$ или менее. Как видно из формул (5.87), в процессе движения пачки ошибок по декодеру слева направо, когда она еще не достигла выхода, ее воздействию подвержены не более двух составных проверок A_i, $i = 1, 2, 3, 4$, и, следовательно, в этот период времени выход порогового элемента всегда равен нулю. В частности, вначале, когда

пачка входит в декодер, она оказывает влияние лишь на проверки A_3 и A_4. После того как пачка достигнет выхода декодера, отличными от нуля шумовыми символами, воздействующими на составные проверки, оказываются лишь символы $e_0^{(1)}$ и $e_0^{(2)}$, поскольку символы $e_b^{(1)}$, $e_b^{(2)}$ и все остальные символы, следующие за ними, равны нулю. Следовательно, ошибка $e_0^{(1)}$ может быть исправлена с помощью порогового декодирования. Аналогично исправляются остальные ошибки $e_1^{(1)}, \ldots, e_{b-1}^{(1)}$.



Фиг. 5.9. Декодер диффузного кода.

5.9.4. Самоортогональные коды

В то время как рассмотренный в последнем примере код представляет собой ортогонализуемый код, коды, предложенные Тонгом [13], являются самоортогональными и строятся следующим образом:

1) Сначала строится код, исправляющий пачки ошибок и имеющий J проверок для каждого информационного символа нулевого блока.

2) Далее этот код с помощью теоремы 5.6 модифицируется таким образом, чтобы он мог исправлять $[J/2]$ и менее независимых ошибок.

Поскольку в действительности построение кодов осуществляется главным образом методом проб и ошибок, то детали построения этих кодов здесь не рассматриваются (с ними можно ознакомиться по оригинальной работе Тонга [13]). Коды, построенные Тонгом, приведены в табл. 5.6.

Тонгом была получена также следующая нижняя граница для кодового ограничения самоортогональных кодов, исправляющих пачки ошибок типа B2 [13].

Теорема 5.7. *Кодовое ограничение самоортогонального кода, исправляющего пачки ошибок типа B2 длиной B блоков, должно*

Таблица 5.6

Таблица самоортогональных диффузных кодов [13]

R	t	n_A	Порождающие многочлены	Минимальное b
1/2	1	$2(3b+1)$	$1+D^b+D^{3b}$	1
	2	$2(4b+2)$	$1+D^b+D^{3b}+D^{4b+1}$	2
	3	$2(5b+5)$	$1+D+D^{b+3}+D^{3b+3}+D^{4b+3}+D^{5b+3}$	4
	4	$2(6b+11)$	$1+D+D^3+D^{b+7}+D^{3b+7}+D^{4b+7}+D^{5b+8}+D^{6b+10}$	8
	5	$2(7b+20)$	$1+D+D^4+D^6+D^{b+12}+D^{3b+12}+D^{4b+12}+D^{5b+13}+D^{6b+15}+D^{7b+19}$	13
2/3	2	$3(8b+4)$	$1+D^b+D^{4b}+D^{8b+3}$ $1+D^{2b}+D^{6b+1}+D^{7b+2}$	3
	3	$3(10b+11)$	$1+D+D^{b+2}+D^{4b+4}+D^{8b+5}+D^{10b+10}$ $1+D^2+D^{2b+2}+D^{6b+4}+D^{7b+4}+D^{9b+7}$	7
	4	$3(12b+27)$	$1+D+D^4+D^{b+7}+D^{4b+11}+D^{8b+12}+D^{9b+13}+D^{11b+24}$ $1+D^2+D^7+D^{2b+7}+D^{6b+11}+D^{7b+11}+D^{10b+20}+D^{12b+2b}$	12
3/4	2	$4(12b+6)$	$1+D^b+D^{6b}+D^{9b+2}$ $1+D^{2b}+D^{11b+3}+D^{12b+5}$ $1+D^{4b}+D^{7b}+D^{8b+1}$	6
	3	$4(15b+13)$	$1+D+D^{2b+3}+D^{11b+6}+D^{12b+7}+D^{15b+12}$ $1+D^2+D^{4b+3}+D^{7b+3}+D^{8b+5}+D^{13b+7}$ $1+D^3+D^b+D^{6b}+D^{9b+2}+D^{14b+8}$	9
4/5	2	$5(16b+9)$	$1+D^b+D^{10b+3}+D^{12b+4}$ $1+D^{2b}+D^{13b+15}+D^{14b+7}$ $1+D^{3b}+D^{7b}+D^{16b+8}$ $1+D^{5b}+D^{8b+1}+D^{9b+2}$	7

удовлетворять следующему неравенству:

$$n_A \geqslant (n_0 - 1)\left(\frac{J}{2} + 2\right) b = \frac{R}{1-R}\left(\frac{J}{2} - 2\right) b. \qquad (5.88)$$

Коды, кодовое ограничение которых достигает этой границы при больших длинах пачек ошибок b, называются квазиоптимальными. В частности, квазиоптимальными являются коды, приведенные в табл. 5.6.

Построенные Тонгом коды имеют скорость $R = (n_0 - 1)/n_0$, однако аналогично можно построить коды со скоростью $R = 1/n_0$. Блоковые коды, исправляющие пачки ошибок и независимые ошибки, были построены также Хсю, Касами и Ченем [43]. Эти коды предназначены для составных каналов; асимптотически достигая нижней границы для длины блоковых кодов, исправляющих одиночные пачки ошибок, они позволяют исправлять несколько независимых ошибок. Указанная выше нижняя граница (5.88) для кодового ограничения самоортогональных кодов больше, чем нижняя граница для кодового ограничения кодов, исправляющих одиночные пачки ошибок. Однако сложность декодирования для кодов, построенных Хсю, Касами и Ченем, того же порядка, что и для БЧХ-кодов, тогда как рассмотренные выше коды допускают более простое пороговое декодирование.

5.10. Равномерные сверточные коды

Другим классом кодов, допускающих пороговое декодирование, является класс равномерных сверточных кодов. Эти коды имеют низкую скорость передачи, но их минимальное расстояние совпадает со средним расстоянием. Обладая приблизительно такой же корректирующей способностью, как и блоковые коды максимальной длины, они тем не менее имеют небольшую глубину распространения ошибок. Однако этот класс кодов является небольшим, поскольку строится лишь для скоростей вида $R = 1/2^m$. Эти коды, впервые построенные Месси [2] для $k_0 = 1$, в дальнейшем были обобщены Робинсоном [36] и на случай $k_0 > 1$. Однако здесь в целях упрощения мы ограничимся рассмотрением только кодов над $GF(2)$ с $k_0 = 1$.

5.10.1. Определение равномерного кода

Определение 5.6. *Двоичным равномерным кодом называется код со скоростью* $R = 1/2^m$, *порождающими многочленами* $G_{(i)}^{(1)}(D)$, $i = 2, 3, \ldots, 2^m$, *которого являются все* $2^m - 1$ *различ-*

ных ненулевых многочленов вида $1 + g_1 D + \ldots + g_m D^m$ *над* $GF(q)$ *степени* m *и менее.*

Например, равномерным кодом является код со скоростью $R = 1/4$, порождаемый многочленами $G_2^{(1)}(D) = 1 + D$, $G_{(3)}^{(1)}(D) = 1 + D^2$, $G_4^{(1)}(D) = 1 + D + D^2$.

5.10.2. Правило ортогонализации

Равномерные коды имеют кодовое ограничение $n_A = (m+1)2^m$, минимальное расстояние $d_{FD} = (m+2)2^{m-1}$ и допускают построение $J = d_{FD} - 1$ проверок, ортогональных относительно $e_u^{(1)}$. Последовательность символов синдрома, образующихся за время $w \leqslant i \leqslant w + m$, определяется матричным соотношением (5.89), где $\bar{H}^{\Delta}$ — матрица, задаваемая формулой (5.90).

Матрицу $\bar{H}^{\Delta}$ будем называть *проверочным параллелограммом.* Подматрица в правой части $\bar{H}^{\Delta}$, выделенная пунктирной линией, является проверочной матрицей H^{Δ}.

$$\begin{bmatrix} s_w^{(2)} \\ s_{w+1}^{(2)} \\ \vdots \\ s_{w+m}^{(2)} \\ s_w^{(3)} \\ \vdots \\ s_{w+m}^{(3)} \\ \vdots \\ s_w^{(2^m)} \\ \vdots \\ s_{w+m}^{(2^m)} \end{bmatrix} = [\bar{H}^{\Delta} : I_{(m+1)\,(2^m-1)}] \begin{bmatrix} \hat{\hat{e}}_{w-m}^{(1)} \\ \vdots \\ \hat{\hat{e}}_{w-1}^{(1)} \\ e_w^{(1)} \\ \vdots \\ e_{w+m}^{(1)} \\ e_w^{(2)} \\ \vdots \\ e_{w+m}^{(2)} \\ e_w^{2^m} \\ \vdots \\ e_{w+m}^{2^m} \end{bmatrix}, \qquad (5.89)$$

$$\bar{H}^{\Delta} = \left[\begin{array}{ccc:ccc} g^{(1)}_{(2),\,m} & \cdots & & g^{(1)}_{(2),\,0} & & \\ & \ddots & & & \ddots & \\ & & \ddots & & & g^{(1)}_{(2),\,0} \\ g^{(1)}_{(3),\,m} & \cdots & & g^{(1)}_{(3),\,0} & & \\ & \ddots & & & \ddots & \\ & & \ddots & & & g^{(1)}_{(3),\,0} \\ \vdots & & & \vdots & & \vdots \\ g^{(1)}_{(2^m),\,m} & \cdots & & g^{(1)}_{(2^m),\,0} & & \\ & \ddots & & & \ddots & \\ & & \ddots & & & g^{(1)}_{(2^m),\,0} \end{array}\right]. \qquad (5.90)$$

Величины $\hat{\hat{e}}^{(1)}_i$, $i = w-m, \ldots, w-1$, в (5.89) будучи суммами $\hat{\hat{e}}^{(1)}_i = e^{(1)}_i + \hat{e}^{(1)}_i$, $i = w-m, \ldots, w-1$, предшествующих шумовых информационных символов $e^{(1)}_i$ и оценок $\hat{e}^{(1)}_i$ последних, равны 0, если $e^{(1)}_i = \hat{e}^{(1)}_i$, т. е. если символ $e^{(1)}_i$ был декодирован верно, равны 1 в противном случае и отражают воздействие на символы синдрома ошибок декодирования.

Месси предложил следующее правило ортогональности равномерных кодов, позволяющее построить систему составных проверок, ортогональных относительно $e^{(1)}_w$:

1. В качестве $2^m - 1$ первых составных проверок выбираются $s^{(j)}_w$, $j = 2, 3, \ldots, 2^m$.

2. Пусть $s^{(j)}_{w+i}$, $i \neq 0$, — символ синдрома, которому соответствует строка $0Q^i$ матрицы H^{Δ} (Q^i — последовательность из i символов 0 и 1, оканчивающаяся символом 1). Тогда найдется по крайней мере один символ $s^{(j')}_{w+i}$, которому в матрице H^{Δ} соответствует строка $1Q^i$. Все суммы $s^{(j)}_{w+i} + s^{(j')}_{w+1}$ относятся к числу искомых составных проверок.

3. В качестве остальных m составных проверок берутся символы $s^{(j)}_{w+i}$, которым соответствуют строки матрицы H^{Δ} вида 11, 101, 10^21, ..., 10^i1, где 0^i — последовательность символов 0 длины i. Заметим, что каждый из шумовых символов $e^{(1)}_{w+i}$, $i = 1, 2, \ldots, m$, контролирует только одну составную проверку.

Пример 5.11. Проверочный параллелограмм равномерного кода со скоростью $R = 1/8$ задается формулой (5.91):

$$\bar{H}^{\Delta} = \left[\begin{array}{ccc:cccc} 0 & 0 & 1 & 1 & & & \\ & 0 & 0 & 1 & 1 & & \\ & & 0 & 0 & 1 & 1 & \\ & & & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & & & \\ & 0 & 1 & 0 & 1 & & \\ & & 0 & 1 & 0 & 1 & \\ & & & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & & & \\ & 0 & 1 & 1 & 1 & & \\ & & 0 & 1 & 1 & 1 & \\ & & & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & & & \\ & 1 & 0 & 0 & 1 & & \\ & & 1 & 0 & 0 & 1 & \\ & & & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & & & \\ & 1 & 0 & 1 & 1 & & \\ & & 1 & 0 & 1 & 1 & \\ & & & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & & & \\ & 1 & 1 & 0 & 1 & & \\ & & 1 & 1 & 0 & 1 & \\ & & & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & & & \\ & 1 & 1 & 1 & 1 & & \\ & & 1 & 1 & 1 & 1 & \\ & & & 1 & 1 & 1 & 1 \end{array}\right]. \qquad (5.91)$$

Метод ортогонализации Месси дает следующую систему ортогональных проверок:

1) $s_{\omega}^{(j)}$, $j = 2, 3, \ldots, 8$;

2) $s_{w+1}^{(2)}+s_{w+1}^{(3)}$, $s_{w+1}^{(4)}+s_{w+1}^{(5)}$, $s_{w+1}^{(6)}+s_{w+1}^{(7)}$,
$s_{w+2}^{(2)}+s_{w+2}^{(4)}$, $s_{w+2}^{(3)}+s_{w+2}^{(5)}$, $s_{w+2}^{(6)}+s_{w+2}^{(8)}$,
$s_{w+3}^{(2)}+s_{w+3}^{(6)}$, $s_{w+3}^{(3)}+s_{w+3}^{(7)}$, $s_{w+3}^{(4)}+s_{w+3}^{(8)}$;

3) $s_{w+1}^{(3)}$, $s_{w+2}^{(7)}$, $s_{w+3}^{(5)}$.

5.10.3. Метод ортогонализации, обеспечивающий небольшую глубину распространения ошибок

Одной из характеристик равномерных кодов является глубина распространения ошибок. Анализируя явление распространения ошибок, Салливан [35] разработал следующий алгоритм ортогонализации, позволяющий минимизировать глубину распространения ошибок:

1) В качестве 2^m-1 первых составных проверок берутся символы синдрома $s_w^{(j)}$, $j=2, 3, \ldots, 2^m$.

2) Пусть $s_{w+i}^{(j)}$ — символ синдрома, которому соответствует строка $R^{m-i}1Q^i$ матрицы $\bar{H}^{\Delta}$, содержащая не менее трех символов 1 (здесь R^{m-i} — последовательность из $m-i$ символов 0 и 1). Тогда среди строк матрицы $\bar{H}^{\Delta}$ всегда можно найти строку вида $R^{m-i}0Q^i$, соответствующую некоторому символу синдрома $s_{w+i}^{(j')}$. Действительно, это всегда можно сделать, поскольку все заканчивающиеся символом 1 строки с номером $i+1$ проверочных параллелограммов образуют совокупность, в которую каждая двоичная последовательность длины $m+1$, содержащая не менее двух единиц, входит ровно один раз. Суммы $s_{w+i}^{(j)}+s_{w+i}^{(j')}$ образуют составные проверки, которые содержат только символ $e_w^{(1)}$.

3) В качестве остальных m составных проверок берутся символы синдрома, которым в матрице $\bar{H}^{\Delta}$ соответствуют строки вида $0^{m-i}1\,0^{i-1}$, $1 \leqslant i < m$.

Пример 5.12. Если применить метод Салливана для ортогонализации кода из примера 5.11, то получим следующую систему составных проверок:

1) $s_w^{(j)}$, $j=2, 3, \ldots, 8$;

2) $s_{w+1}^{(3)}+s_{w+1}^{(4)}$, $s_{w+1}^{(5)}+s_{w+1}^{(6)}$, $s_{w+1}^{(7)}+s_{w+1}^{(8)}$,
$s_{w+2}^{(2)}+s_{w+2}^{(4)}$, $s_{w+2}^{(5)}+s_{w+2}^{(7)}$, $s_{w+2}^{(6)}+s_{w+2}^{(8)}$,
$s_{w+3}^{(2)}+s_{w+3}^{(6)}$, $s_{w+3}^{(3)}+s_{w+3}^{(7)}$, $s_{w+3}^{(4)}+s_{w+3}^{(8)}$;

3) $s_{w+1}^{(2)}$, $s_{w+2}^{(3)}$, $s_{w+3}^{(5)}$.

Метод ортогонализации Салливана является частным случаем метода ортогонализации Месси, но он выгодно отли-

чается от последнего следующим. Использование метода ортогонализации Месси в примере 5.11 приводит, например, к тому, что от $\hat{\hat{e}}^{(1)}_{w-1}$ оказываются зависящими восемь составных проверок. Но в этом случае возникновение ошибки декодирования, когда $\hat{\hat{e}}^{(1)}_{w-1} \neq 0$, эквивалентно появлению пачки ошибок длины 8, что резко ухудшает корректирующую способность кода по отношению к символу $e^{(1)}_w$. В то же время при использовании метода ортогонализации Салливана ни одна из составных проверок, полученных на втором и третьем шагах, не содержит ни один из символов $\hat{\hat{e}}^{(1)}_i$, $i = w-m, \ldots, w-1$. Поэтому с точки зрения уменьшения глубины распространения ошибок метод Салливана является оптимальным методом ортогонализации.

Поскольку при использовании метода ортогонализации Салливана величины $\hat{\hat{e}}^{(1)}_{w-m}, \ldots, \hat{\hat{e}}^{(1)}_{w-1}$ контролируются лишь проверками $s^{(j)}_w$, $j = 2, 3, \ldots, 2^m$, то их воздействие на составные проверки может быть задано следующим простым соотношением:

$$\begin{bmatrix} s^{(2)}_w \\ s^{(3)}_w \\ \cdot \\ \cdot \\ s^{(2^m)}_w \end{bmatrix} = \bar{\Lambda} \begin{bmatrix} \hat{\hat{e}}^{(1)}_{w-m} \\ \cdot \\ \cdot \\ \cdot \\ \hat{\hat{e}}^{(1)}_{w-1} \end{bmatrix} + \begin{bmatrix} e^{(1)}_w \\ e^{(1)}_w \\ \cdot \\ \cdot \\ e^{(1)}_w \end{bmatrix} + \begin{bmatrix} e^{(2)}_w \\ e^{(3)}_w \\ \cdot \\ \cdot \\ e^{(2^m)}_w \end{bmatrix}, \qquad (5.92)$$

где $\bar{\Lambda}$ — матрица размера $(2^m - 1) \times m$, строками которой являются всевозможные ненулевые последовательности длины $m+1$.

Будем говорить, что декодер в момент времени $w+m$ находится в нормальном рабочем состоянии, если $\hat{\hat{e}}^{(1)}_{w-m} = \hat{\hat{e}}^{(1)}_{w-m+1} = \ldots = \hat{\hat{e}}^{(1)}_{w-1} = 0$. Пусть E_w — множество ошибок, контролируемых составными проверками, т. е. $E_w = \{e^{(j)}_i\colon\ i = w, \ldots, w+m, j = 1, 2, \ldots, 2^m\}$.

Теорема 5.8. *Пусть при использовании равномерного сверточного кода и оптимального правила ортогонализации некоторая некорректируемая совокупность ошибок в канале приводит к ошибочному декодированию символа* $e^{(1)}_i$. *Пусть* w — *минимальное целое число, такое, что ни одно из множеств* E_w, $E_{w+1}, \ldots, E_{w+m-1}$ *не содержит* $Y(m)$ *или более символов* 1

$(w > i)$, *где*

$$Y(m) = \begin{cases} 0, & m = 1, \\ m2^{m-2} - 1, & m \geqslant 2. \end{cases}$$

Тогда декодер к моменту времени $w + 2m$ *перейдет в нормальное рабочее состояние.*

Доказательство. Сначала предположим, что $e_w^{(1)} = 0$ и ни один из элементов E_w не равен 1. В этом случае в правой части (5.92) ненулевым является только первый вектор, а следовательно, ровно 2^{m-1} символов множеств $\{s_w^{(j)},\ j = 2, 3, \ldots, 2^m\}$ будут равны единице [1]). Допустим, что в E_w равны 1 не более $Y(m)$ символов $(e_w^{(1)} = 0)$. Тогда изменить свое значение с 0 на 1 могут самое большее $Y(m)$ составных проверок. Следовательно, общее число составных проверок, принимающих значение 1, не больше, чем целая часть $[(2 + m)2^{m-1} - 1]/2$, и решение о $e_w^{(1)}$ принимается правильно.

Далее предположим, что $e_w^{(1)} = 1$, и все элементы E_w, за исключением $e_w^{(1)}$, равны нулю. Тогда среди элементов множества $\{s_w^{(j)},\ j = 2, 3, \ldots, 2^m\}$ будут равны 0 самое большее 2^{m-1} элементов. Если допустить, что среди символов E_w равны 1 не больше чем $Y(m)$, то число символов E_w, равных 1 и отличных от $e_w^{(1)}$, строго меньше чем $Y(m)$, а следовательно, строго меньше чем $Y(m)$ составных проверок изменяют свое значение с 1 на 0. В этом случае не менее $[(2 + m)2^{m-1} - 1]/2$ составных проверок по-прежнему будут оставаться равными единице и решение о $e_w^{(1)}$ опять будет принято правильно.

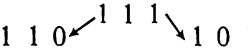
Из доказанной теоремы следует, в частности, что декодер кода из примера 5.11 принимает правильное решение о $e_w^{(1)}$ при наличии в E_w не более девяти символов 1, если ошибок декодирования раньше не было, и при наличии в E_w не более пяти символов 1, если перед этим произошла ошибка декодирования.

5.10.4. Перфорированные равномерные коды

Равномерные коды имеют очень высокую корректирующую способность при низкой скорости передачи. Скорость передачи можно повысить путем выбрасывания некоторых проверочных символов, ухудшив при этом несколько корректирующую способность. Этот метод построения кодов был использован Робинсоном [36]. Рассмотрим *решетку*, узлами которой являются дво-

[1]) Это связано с тем, что сумма любого ненулевого числа столбцов матрицы $\tilde{\Lambda}$ всегда является вектором веса 2^{m-1}. — *Прим. перев.*

ичные векторы, соответствующие различным проверочным соотношениям (порождающим многочленам) и соединенные определенным образом друг с другом стрелками. Например, решетка равномерного кода с параметрами $n_0 = 4$, $k_0 = 1$ и $m = 2$ имеет вид



Узел 111 этой решетки представляет следующие три проверочных соотношения:

$$s_w^{(2)} = e_{w-2}^{(1)} + e_{w-1}^{(1)} + e_w^{(1)},$$

$$s_{w+1}^{(2)} = e_{w-1}^{(1)} + e_w^{(1)} + e_{w+1}^{(1)},$$

$$s_{w+2}^{(2)} = e_w^{(1)} + e_{w+1}^{(1)} + e_{w+2}^{(1)}.$$

Стрелка, соединяющая узлы 111 и 101, указывает шаг процесса ортогонализации кода, позволяющий путем сложения символа синдрома

$$s_{w+1}^{(3)} = e_{w-1}^{(1)} + e_{w+1}^{(1)}$$

с $s_{w+1}^{(2)}$ получить такую составную проверку, что $s_{w+1}^{(2)} + s_{w+1}^{(3)} + e_w^{(1)} = 0$. Аналогично стрелка, соединяющая узлы 110 и 111, указывает шаг процесса ортогонализации, позволяющий путем сложения символа синдрома

$$s_{w-1}^{(4)} = e_{w-2}^{(1)} + e_{w-1}^{(1)}$$

с $s_w^{(2)}$ получить такую проверку, что

$$s_{w-1}^{(4)} + s_w^{(2)} + s_w^{(1)} = 0.$$

Последняя стрелка указывает шаг ортогонализации, позволяющий путем сложения символа синдрома

$$s_{w+2}^{(4)} = e_{w+1}^{(1)} + e_{w+2}^{(1)}$$

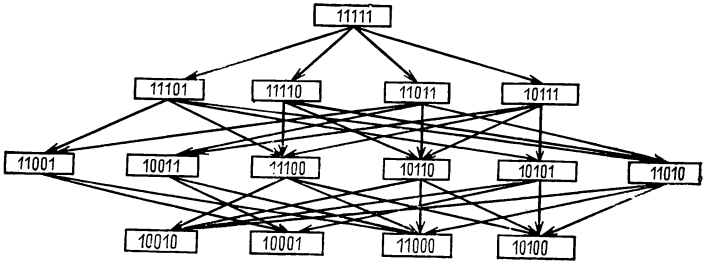
с $s_{w+2}^{(2)}$ получить составную проверку, такую, что

$$s_{w+2}^{(2)} + s_{w+2}^{(4)} + e_w^{(1)} = 0.$$

Общее правило построения решетки состоит в следующем: вектор V j-го яруса соединяется с вектором W $(j+1)$-го яруса, если покомпонентная сумма по модулю 2 этих векторов или

одного из них с некоторым сдвигом другого является вектором веса 1. Вектор, расположенный на самом высоком ярусе, имеет вес $k_0(m+1)$, а векторы, расположенные на самом нижнем ярусе, имеют вес 2.

Рассмотрим, каким образом можно повысить скорость передачи за счет выбрасывания проверочных символов. На фиг. 5.10 показана решетка кода с параметрами $m=4$ и $k_0=1$. Удалим вектор 11111, расположенный в четвертом ярусе. Заметим, что в этом случае оставшимся узлам решетки соответствуют символы синдрома, ортогонализация которых осуществляется без использования символов синдрома, соответствующих вектору 11111, а следовательно, ортогонализацию полученного кода можно провести точно так же, как и исходного кода.



Фиг. 5.10. Решетчатая структура кода с $m=4$ и $k_0=1$.

Поскольку выброшенный узел имел вес 5, то минимальное расстояние кода уменьшается с 48 до 43; скорость передачи при этом возрастает с $^1/_{16}$ до $^1/_{15}$. Далее без нарушения процесса ортогонализации можно выбросить произвольный вектор, расположенный в 3-м ярусе. После того как будут выброшены все векторы 3-го яруса, можно перейти к выбрасыванию векторов 2-го яруса.

Однако можно воспользоваться другой процедурой выбрасывания, позволяющей получить коды с большим минимальным расстоянием. Эта процедура заключается в следующем. При выбрасывании любого узла решетки число проверочных символов кода уменьшается на единицу, а минимальное расстояние — на вес элемента, соответствующего выброшенному узлу. Если обратиться к процессу ортогонализации, когда в результате сложения двух символов синдромов образуются составные проверки, то можно заметить, что узел решетки можно выбросить, если он не связан с оставшимися узлами более высоких ярусов. Рассматривая каждый вектор как двоичную запись некоторого десятичного числа (старшим разрядом будем считать самый правый символ), упорядочим узлы решетки в порядке убывания

этих десятичных чисел (номеров). При построении кода узлы решетки будем выбрасывать в порядке убывания их номеров. Это всегда можно делать, поскольку вектор j-го яруса, имеющий номер b, не связан ни с одним из векторов $(i+1)$-го яруса, имеющим номер, меньший чем b. Например, с помощью этого алгоритма выбрасывания может быть получен код с параметрами $R = 1/11$, $d = 28$, $n_A = 55$ (при этом из решетки, изображенной на фиг. 5.10, выбрасываются узлы 11111, 10111, 11011, 11101). Получающиеся таким образом коды называются *перфорированными равномерными кодами.*

Задачи

1. Построить кодер и декодер кода из табл. 5.1 со скоростью передачи $R = {}^2/_3$ и $J = 6$ ортогональными проверками.

2. а) Показать, что совокупности целых чисел

$$\{2,\ 10,\ 7,\ 6,\ 5,\ 3,\ 1\},$$

$$\{3,\ 6,\ 8,\ 2,\ 13,\ 7,\ 4,\ 1\}$$

задают порождающие многочлены самоортогональных кодов со скоростью $R = {}^1/_2$ и соответственно $J = 8$ и $J = 9$ ортогональными проверками.

б) Сравнить кодовое ограничение указанного в п. «а» кода с параметрами $R = {}^1/_2$ и $J = 8$ с кодовым ограничением соответствующего кода из табл. 5.1.

(Приведенные выше две совокупности целых чисел были недавно найдены Робинсоном [45].)

3. Сверточные коды, исправляющие одиночные ошибки [6]. Эти коды имеют параметры

$$n_0 = 2^{m-1},$$

$$k_0 = n_0 - 1,$$

$$d = 3$$

и строятся следующим образом.

В качестве первых n_0 столбцов матрицы H, т. е. столбцов матрицы B, берутся 2^{m-1} двоичных векторов длины m, первая компонента которых равна 1 (n_0-й столбец матрицы B_0, соответствующий проверочному символу, должен быть вектором, все компоненты которого, за исключением первой, равны 0).

а) Показать, что определяемый такой матрицей $(m2^{m-1}, m(2^{m-1}-1))$-код исправляет одиночные ошибки.

б) Построить матрицу B_0 в случае $n_0 = 2^{m-1} = 8$.

4. Коды, исправляющие две независимые ошибки [6]. Пусть α — примитивный элемент поля $GF(2^m)$, H_1 и H_2 — матрицы размера $m \times (2^m - 1)$:

$$H_1 = [\alpha^{2^m-2}, \alpha^{2^m-3}, \ldots, \alpha^1, \alpha^0],$$

$$H_2 = [\alpha^{3(2^m-2)}, \alpha^{3(2^m-3)}, \ldots, \alpha^3, \alpha^0],$$

1 — вектор-строка, все $2^m - 1$ компонент которого равны 1, а **0** — вектор-строка, все $2^m - 1$ компонент которого равны 0.

Показать, что сверточный код, который задается матрицей

$$H = \begin{bmatrix} \mathbf{1} & \\ H_1 & \\ \mathbf{0} & \mathbf{1} \\ H_2 & H_2 \end{bmatrix},$$

может исправлять любые две независимые ошибки.

5. БПМ-коды [5, 7, 8].

Берлекэмп, Препарата и Месси построили коды, исправляющие пачки ошибок типа B2 длины $b = n_0$. Рассмотрим случай $n_0 = 6$. Пусть B_0 — матрица следующего вида:

$$B_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A \\ 0 & 0 & 0 & 0 & C & B \\ 0 & 0 & 0 & F & E & D \\ 0 & 0 & J & I & H & G \\ 0 & O & N & M & L & K \end{bmatrix},$$

где $A, B, C, \ldots$ — двоичные переменные, которые нужно выбрать. Проверочной матрицей этого кода является матрица (5.93). Путем сложения столбцов 0-го блока со столбцами последующих блоков можно превратить в нули все единичные символы, ле-

жащие выше горизонтальной линии.

0	1	2	3	4	5
10 0 0 0 0					
01 0 0 0 0	10000 0				
00 1 0 0 1	01000 0	100000			
00 0 1 0 0	00100 0	010000	100000		
00 0 0 1 0	00010 0	001000	010000	100000	
00 0 0 0 1	00001 0	000100	001000	010000	100000
00 0 0 0 0	00000 0	000010	000100	001000	010000
00 0 0 0 *A*	00000 0	000001	000010	000100	001000
00 0 0 *C B*	00000 *A*	000000	000001	000010	000100
00 0 *F E D*	0000 *C B*	0000 0 *A*	000000	000001	000010
00 *J I H G*	000 *F E D*	0000 *C B*	00000 *A*	000000	000001
0 *O N M L K*	00 *J I H G*	000 *F E D*	0000 *C B*	00000 *A*	000000

(5.93)

а) Пусть M_1, M_2, M_3, M_4, M_5 — подблоки матрицы, получающейся из (5.93) в результате выполнения указанных выше операций, расположенные ниже горизонтальной линии и соответствующие блокам 1—5 матрицы (5.93). Найти M_1, M_2, M_3, M_4, M_5.

б) Найти необходимое и достаточное условие того, что этот код исправляет пачки ошибок типа B2.

в) В результате указанных в п. «а» операций неизвестный символ A появляется в левой верхней подматрице размера 2×2 матрицы M_5:

$$\begin{matrix} 0 & 1 \\ A & 0 \end{matrix}$$

Если выбрать $A = 1$, то эта подматрица будет невырожденной. Неизвестный символ B входит в подматрицу

$$\begin{matrix} 0 & 1 & 0 \\ A & 0 & 1 \\ B & 0 & 0 \end{matrix} = \begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ B & 0 & 0 \end{matrix}$$

Если выбрать $B = 1$, то эта подматрица также будет невырожденной. Аналогично найти остальные символы C, D, ...

г) Определить защитный интервал этого кода.

д) Пусть этот код используется для исправления пачек ошибок типа B1. Определить необходимый защитный интервал

и сравнить его с защитным интервалом соответствующего кода Ивадари.

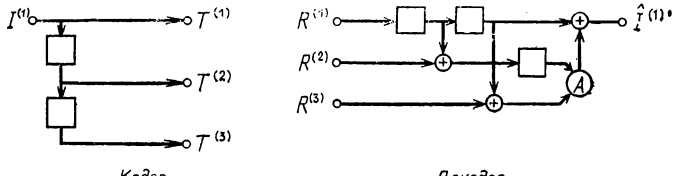
6. а) Построить матрицу B_0 кода Хегельбергера с параметрами $n_0 = 4$, $b = 4$.

б) Начертить схемы кодера и декодера для этого кода.

в) Сравнить защитный интервал этого кода с границей (5.72).

7. а) Найти порождающий многочлен кода Ивадари типа II со скоростью $R = 3/4$, исправляющего пачки ошибок длины $b = 8$.

б) Построить кодер и декодер для этого кода.



Кодер Декодер

Фиг. 5.11. Кодер и декодер кода, исправляющего пачки ошибок длины $b = 3$ (скорость $R = 1/3$).

в) Рассмотреть работу декодера при коррекции следующей пачки ошибок длины [8]:

$$E^{(1)}(D) = e_1^{(1)}D + e_2^{(1)}D^2,$$

$$E^{(2)}(D) = e_0^{(2)} + e_1^{(2)}D,$$

$$E^{(3)}(D) = e_0^{(3)} + e_1^{(3)}D,$$

$$E^{(4)}(D) = e_0^{(4)} + e_1^{(4)}D.$$

8. Известны коды [12], исправляющие пачки ошибок типа B1, которые задаются синдромами

$$s^{(n_0)}(D) = E^{(n_0)}\underbrace{00\ldots0}_{\beta+n_0-2}E^{(n_0-1)}\underbrace{0\ldots0}_{n_0-2}E^{(n_0-1)}E^{(n_0-2)}\underbrace{0\ldots0}_{n_0-3}E^{(n_0-2)}\ldots$$

$$\ldots E^{(1)}E^{(1)}$$

и

$$s^{(n_0)}(D) =$$

$$= E^{(n_0)}\underbrace{00\ldots0}_{n_0-1}E^{(n_0-1)}E^{(n_0-2)}\ldots E^{(1)}\underbrace{00\ldots0}_{\beta+n_0-1}E^{(n_0-1)}0E^{(n_0-2)}0E^{(n_0-3)}0\ldots$$

$$\ldots 0E^{(1)}.$$

а) Рассмотреть работу декодера в случае $n_0 = 3$, $\beta = 2$.

б) Определить необходимый защитный интервал и число ячеек регистров сдвига, необходимых для построения кодера и декодера.

9. Коды, исправляющие пачки ошибок со скоростью $R = 1/n_0$ [10].

а) Показать, что изображенные на фиг. 5.11 кодер и декодер позволяют исправлять пачки ошибок длины $b = 3$ и менее.

б) Обобщить указанные в п. «а» методы кодирования и декодирования на случай произвольных кодов со скоростью $R = 1/n_0$.

10. а) Выписать порождающие многочлены равномерного кода со скоростью $R = 1/16$.

б) Провести ортогонализацию этого кода методом Салливана.

6. Сверточные коды
II. Последовательное декодирование

Последовательное декодирование было предложено Возенкрафтом [1], и, как уже указывалось выше, на начальном этапе исследования в этой области велись главным образом в Массачусетском технологическом институте. Кодирование и декодирование блоковых кодов и рассматривавшихся до сих пор сверточных кодов основано главным образом на использовании алгебраических методов. Основные достоинства метода последовательного декодирования связаны с тем, что при этом методе число операций, необходимых для декодирования одного символа, является случайной величиной в отличие от того, что имеет место при алгебраических методах.

Если простота алгоритмов порогового декодирования и, следовательно, простота их реализации стимулировали исследования сравнительно коротких кодов и повышение их корректирующих способностей, то проводившиеся параллельно исследования методов последовательного декодирования имели одной из своих главных целей практическую реализацию теоремы Шеннона для канала с шумами [2]. При доказательстве теоремы кодирования Шеннон и Галлагер [3] использовали декодирование по максимуму правдоподобия. Алгоритм декодирования по максимуму правдоподобия применительно к сверточным кодам предложен Витерби [4]. Однако при декодировании по максимуму правдоподобия сложность устройств с увеличением длины кода растет экспоненциально. Достоинством методов последовательного декодирования является то, что они требуют для своей реализации приемлемого по стоимости оборудования и при этом позволяют достичь приблизительно той же вероятности ошибки, что и методы оптимального декодирования по максимуму правдоподобия, стоимость реализации которых является очень высокой. Это связано с тем, что при последовательном декодировании число операций, которые должен выполнить декодер для того, чтобы декодировать один символ, изменяется в зависимости от уровня шумов в канале. Число операций при последовательном декодировании является функцией скорости передачи и шумов в канале. При всех скоростях передачи, меньших определенной скорости $R_{\text{выч}}$, среднее число операций при декодировании оказывается небольшим. Последовательный декодер состоит из логической схемы, позволяющей проводить вычисления со средней скоростью, в несколько раз большей ско-

рости передачи символов, и буферного запоминающего устройства, предназначенного для хранения поступающих данных при повышении уровня шумов в канале. В случае если число возникших в канале ошибок превысит корректирующую способность кода, при последовательном декодировании и других методах декодирования могут возникать ошибки. Однако при последовательном декодировании ошибки могут возникать также и из-за того, что оказываются занятыми все ячейки памяти буферного запоминающего устройства, т. е., другими словами, из-за переполнения буфера. Конечно, если вероятность возникновения ошибки из-за переполнения буфера оказывается значительно больше, чем вероятность возникновения ошибки из-за невозможности исправить или обнаружить ошибку, то именно она и будет определять характеристики алгоритма последовательного декодирования.

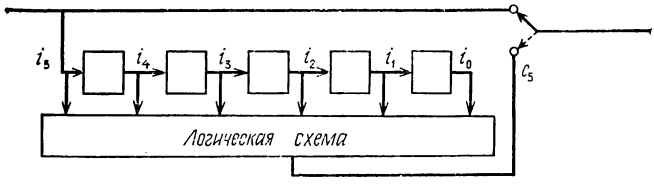
Число операций, необходимых для декодирования одного символа, при последовательном декодировании является случайной величиной, и поэтому важным методом анализа последовательного декодирования является метод, использованный Галлагером [3], а также Шенноном, Галлагером и Берлекэмпом [5] для получения границы случайного кодирования. Однако даже при использовании этого эффективного метода провести полный теоретический анализ не удается, и поэтому важным методом анализа последовательного декодирования остается его моделирование.

6.1. Древовидные коды и принцип последовательного декодирования

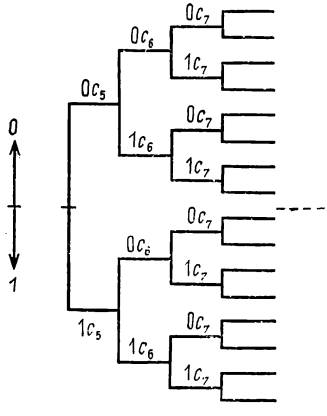
6.1.1. Древовидные коды

Рассмотрим кодер сверточного кода, схема которого изображена на фиг. 6.1. Последовательность информационных символов поступает в регистр сдвига, состоящий из пяти ячеек; сдвиг регистра производится со скоростью R бит/с. Через каждые $1/R$ с на выход кодера поступают два символа, один из которых является информационным, а второй — проверочным. Каждый проверочный символ в данном случае является суммой по модулю 2 текущего и пяти предыдущих информационных символов.

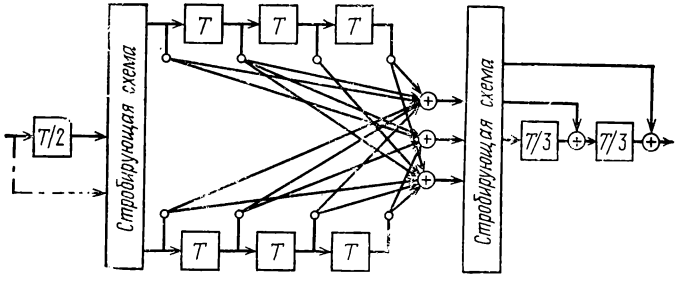
Одним из способов представления сверточных кодов является представление их в виде кодовых деревьев. На фиг. 6.2 показано кодовое дерево сверточного кода, порождаемого описанным выше кодером. Символы (i_j, c_j), показанные на этом дереве, являются соответственно информационным и проверочным



Фиг. 6.1. Пример сверточного кодера.



Фиг. 6.2. Пример древовидного кода.



Фиг. 6.3. Пример кодера с $R = 2/3$.

символами на выходе кодера в момент времени j. Верхнее ребро, выходящее из каждого узла на фиг. 6.2, соответствует значению 0 информационного символа, а нижнее ребро — значению 1. Если последовательность информационных символов задана, то полностью определенными оказываются и проверочные символы, и, следовательно, они могут быть указаны на соответствующих ребрах вместе с информационными символами. Структура древовидного кода такова, что каждой передаваемой информационной последовательности соответствует определенный *путь* на кодовом дереве.

Рассмотрим кодер, схема которого изображена на фиг. 6.3. Этот кодер построен таким образом, что после того как первый информационный символ поступит в верхний регистр сдвига, а второй информационный символ — в нижний регистр сдвига, с помощью трех сумматоров по модулю 2 формируются три символа канала. Каждая последующая пара информационных символов также направляется в верхний и нижний регистры сдвига, после чего осуществляется вычисление очередных выходных символов. В данном случае каждым двум информационным символам сопоставляются три символа канала и, следовательно, рассматриваемый код имеет скорость $R = {}^2/_3$. Так как каждый информационный символ оказывает влияние на 12 последующих выходных символов, то кодовое ограничение этого кода равно 12.

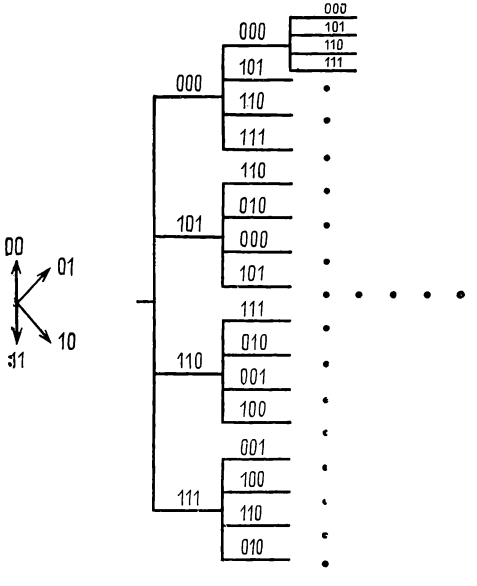
Аналогично, используя k_0 m-разрядных регистров сдвига и n_0 сумматоров, можно построить кодер кода со скоростью $R = k_0/n_0$ и кодовым ограничением $n_A = n_0(m+1)$. В двоичном случае из каждого узла кодового дерева, соответствующего такому коду, выходит 2^{k_0} ребер, каждому из которых сопоставлено n_0 двоичных символов. Кодовое дерево кода, кодер которого изображен на фиг. 6.3, показано на фиг. 6.4. Коды, представляемые в виде кодового дерева, называются *древовидными*.

Вообще говоря, древовидными кодами могут быть названы произвольные коды, имеющие древовидную структуру, подобную показанной на фиг. 6.2 или фиг. 6.4. При таком широком определении древовидных кодов коды, порождаемые сверточными кодерами, являются лишь подмножеством множества древовидных кодов.

6.1.2. Принцип последовательного декодирования

Предположим, что задан некоторый древовидный код и попытаемся по принятой последовательности определить передававшуюся последовательность. Для этого сначала будем передвигаться поочередно вдоль каждого пути кодового дерева, сравнивая принятую последовательность с последовательностью,

соответствующей пути, по которому мы движемся. Если при этом удается обнаружить некоторый путь, которому соответствует последовательность, почти совпадающая с принятой последовательностью, то естественно считать, что эта последовательность и передавалась. Метод декодирования, предложенный Возенкрафтом, предусматривает вначале поиск на кодовом дереве пути из $(m+1)$-го ребра, соответствующая которому кодовая последовательность находится на каком-то определенном расстоянии от переданной последовательности, и далее декодирование первого информационного символа (первым передан-



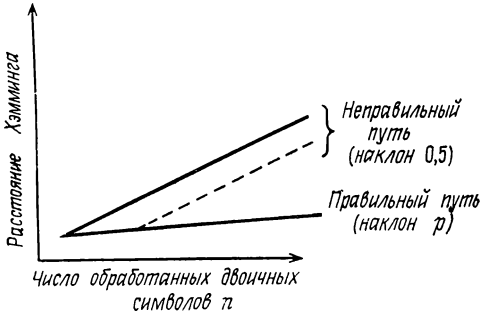
Фиг. 6.4. Пример древовидного кода скоростью $R = {}^2/_3$.

ным информационным символом считается первый информационный символ первого ребра найденного пути). Поскольку далее декодирование второго и последующих информационных символов осуществляется точно так же, как и первого информационного символа, то этот алгоритм был назван алгоритмом последовательного декодирования.

Одним из критериев близости пути на кодовом дереве к принятой последовательности является расстояние Хэмминга между ними; если расстояние Хэмминга между последовательностями длины n не превышает определенный порог l_n, то последовательности можно считать совпадающими. В случае двоичного симметричного канала порог l_n выбирается таким образом, чтобы вероятность того, что расстояние Хэмминга между

передававшейся и принятой последовательностями было больше чем l_n, не превышала величины 2^{-L}, где L — некоторая заданная постоянная. При этом вероятность того, что расстояние Хэмминга между правильным путем дерева и принятой последовательностью больше чем l_n, при больших L оказывается чрезвычайно малой.

Предположим, что этот метод декодирования используется в двоичном симметричном канале с вероятностью перехода p. Тогда, поскольку в этом случае в канале ошибки в различных символах возникают независимо с вероятностью перехода p, то расстояние Хэмминга, вычисляемое вдоль правильного пути, как показано на фиг. 6.5, почти всегда будет возрастать со скоростью p вместе с числом обработанных символов. В то же время



Фиг. 6.5. Число декодированных символов и расстояние Хэмминга для последовательного декодера.

расстояние Хэмминга, вычисляемое вдоль любого другого пути кодового дерева, будет возрастать со скоростью $1/2$, поскольку различные пути кодового дерева отличаются друг от друга приблизительно в половине символов. Таким образом, расстояния Хэмминга, вычисленные вдоль правильного и ошибочного пути кодового дерева, оказываются различными. При этом, даже если на начальном этапе декодирования путь кодового дерева был выбран неправильно, то через некоторое время ошибка будет обнаружена, поиск правильного пути будет продолжен и в конце концов правильный путь будет найден.

Этот алгоритм последовательного декодирования был обобщен Рейффеном [6] на случай произвольного дискретного канала без памяти. Обобщение было сделано путем замены расстояния Хэмминга на другую функцию, являющуюся обобщением расстояния Хэмминга.

Дискретный канал без памяти с входным алфавитом из K символов и выходным алфавитом из J символов определяется совокупностью переходных вероятностей $p(v_j|u_i)$, $i \leqslant i \leqslant K$,

$1 \leqslant j \leqslant J$, где $p(v_j|u_i)$ — вероятность того, что при передаче входного символа u_i на выходе канала будет принят символ v_j. Расстояние между путем кодового дерева (т. е. $u_n = (u_1, u_2, \ldots, u_n)$, содержащим n символов, и принятой последовательностью $\bar{v}_n = (v_1, v_2, \ldots, v_n)$ определяется равенством

$$I(\bar{u}_n, \bar{v}_n) = \sum_{i=1}^{n} \log_2 \frac{p(v_i|u_i)}{Q(v_i)}, \tag{6.1}$$

где

$$Q(v_i) = \sum_{i-1}^{n} q_i p(v_i|u_i). \tag{6.2}$$

Величина $Q(v_i)$ представляет собой вероятность появления на выходе канала символа v_i в случае, если на входе канала символ u_i появляется с вероятностью q_i, $1 \leqslant i \leqslant K$. Расстояние, определяемое формулой (6.1), является количеством информации между последовательностями $\bar{u}_n$ и $\bar{v}_n$ в случае, когда пары (u_i, v_i) статистически независимы и символы u_i на входе канала появляются с вероятностями q_i, $1 \leqslant i \leqslant K$. Если правую часть (6.1) представить в виде логарифма произведения, то можно заметить, что величина $I(\bar{u}_n, \bar{v}_n)$ определяется вероятностью $p(\bar{v}_n|\bar{u}_n)$, которая в случае двоичного симметричного канала является монотонно убывающей функцией расстояния Хэмминга между последовательностями $\bar{u}_n$ и $\bar{v}_n$. Следовательно, в случае двоичного симметричного канала введенное выше расстояние эквивалентно расстоянию Хэмминга. Однако поскольку в случае произвольного дискретного канала без памяти введенная функция не может быть интерпретирована в терминах расстояния Хэмминга, то будем называть ее в дальнейшем *ценой пути*.

Алгоритм последовательного декодирования, предложенный Фано [23], предусматривает вычисление и использование при декодировании наряду с ценой пути также некоторых порогов. Этот алгоритм является алгоритмом того же типа, что и алгоритм Возенкрафта, но если среднее число операций, необходимых для декодирования одного символа, в случае использования алгоритма Возенкрафта при скоростях передачи, меньших определенной скорости $R_{\text{выч}}$, растет как небольшая степень кодового ограничения, то для алгоритма Фано это среднее число в тех же условиях не зависит от длины кодового ограничения. Это упрощает анализ алгоритма декодирования Фано. Кроме того, достоинством алгоритма Фано является также то, что он допускает более простую техническую реализацию, чем алгоритм Возенкрафта. В силу этого приблизительно с 1964 г. исследования алгоритмов последовательного декодирования в значительной степени были связаны с исследованием алгоритма

Фано. По тем же причинам алгоритм Фано будет детально рассмотрен в данной книге. В последние годы независимо Джелинек [7] и Зигангиров [8] предложили алгоритм последовательного декодирования, часто называемый *стек-алгоритмом*. Нельзя сказать, что этот алгоритм во всех отношениях лучше алгоритма Фано, но так как его описание значительно проще описания алгоритма Фано и поскольку он хорошо иллюстрирует принцип последовательного декодирования, то этот алгоритм также будет кратко рассмотрен ниже.

6.2. Алгоритм Фано

При декодировании сверточных кодов с помощью алгоритма Фано точно так же, как и в случае алгоритма Возенкрафта, декодер всякий раз вычисляет цену пути для принятой последовательности и одного из путей на кодовом дереве. В случае, если найденная цена пути превышает определенный порог, то осуществляется декодирование очередного информационного символа. Если же цена пути оказывается ниже порога, то декодер полагает, что он попал на ошибочный путь, и начинает поиск правильного пути. Осуществляя декодирование таким образом, декодер Фано постепенно приближается к правильному пути кодового дерева, а следовательно, и к правильному декодированию информационных символов. Прежде чем перейти к описанию алгоритма, рассмотрим, что в данном случае понимается под ценой пути.

6.2.1. Цена пути

Будем предполагать, что для каждого пути $\bar{u}_n$ на кодовом дереве декодер может вычислить вероятность $p(\bar{u}_n|\bar{v}_n)$ того, что передавалась последовательность $\bar{u}_n$ при условии, что принята последовательность $\bar{v}_n$. Здесь $\bar{u}_n$ и $\bar{v}_n$ — последовательности длины n:

$$\begin{aligned} \bar{u}_n &= (u_1, u_2, \ldots, u_n), \\ \bar{v}_n &= (v_1, v_2, \ldots, v_n). \end{aligned} \tag{6.3}$$

Рассмотрим алгоритм, который, сравнивая эту вероятность $p(\bar{u}_n|\bar{v}_n)$ с некоторым порогом, принимает решение о том, является ли данная ветвь правильной или неправильной. Вероятность $p(\bar{u}_n|\bar{v}_n)$ можно представить в следующем виде:

$$p(\bar{u}_n|\bar{v}_n) = \frac{p(\bar{u}_n, \bar{v}_n)}{Q(\bar{v}_n)} = \frac{Q(\bar{u}_n)\, p(\bar{v}_n|\bar{u}_n)}{Q(\bar{v}_n)}. \tag{6.4}$$

Если предположить, что канал является каналом без памяти, то

$$p(\bar{v}_n|\bar{u}_n) = \prod_{i=1}^{n} p(v_i|u_i). \tag{6.5}$$

Для простоты рассмотрим случай двоичных кодов и, кроме того, предположим, что все пути кодового дерева порождаются с равными вероятностями. В этом случае

$$Q(\bar{u}_n) = 2^{-Rn},$$

где R — скорость передачи, выраженная в битах на символ канала. Таким образом,

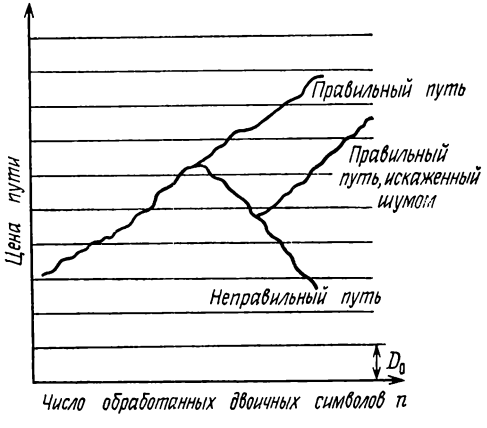
$$p(\bar{u}_n \mid \bar{v}_n) = 2^{-Rn} \prod_{i=1}^{n} \frac{p(v_i \mid u_i)}{Q(v_i)}. \tag{6.6}$$

Если взять логарифм обеих частей равенства (6.6), то получим

$$\log p(\bar{u}_n \mid \bar{v}_n) = \sum_{i=1}^{n} \left[\log \frac{p(v_i \mid u_i)}{Q(v_i)} - R \right] =$$

$$= \sum_{i=1}^{n} \log \frac{p(v_i \mid u_i)}{Q(v_i)} - nR. \tag{6.7}$$

Заметим, что сумма по i в правой части этого равенства является взаимной информацией $I(\bar{u}_n, \bar{v}_n)$ между последователь-



Фиг. 6.6. Изменение цен путей для алгоритма Фано.

ностями $\bar{u}_n$ и $\bar{v}_n$. Следовательно, при подходящем выборе скорости передачи R среднее значение (6.7) для правильного пути будет положительным, а для неправильного пути — отрицательным. Более того, вдоль правильного пути это среднее значение (6.7) является возрастающей функцией n. Поэтому в общем случае определим цену пути следующим образом:

$$\gamma(\bar{u}_n, \bar{v}_n) = I(\bar{u}_n, \bar{v}_n) - nU, \tag{6.8}$$

где U — коэффициент сноса, который будет определен ниже. В качестве порогов, с которыми сравниваются цены путей, рассмотрим совокупность прямых, параллельных оси абсцисс и расположенных друг от друга на расстоянии D_0 (эта совокупность порогов удобна с точки зрения практической реализации алгоритма). При подходящем выборе величины U цены правильного и неправильного путей изменяются обычно так, как показано на фиг. 6.6. Рассматриваемый здесь алгоритм Фано в конце концов позволяет найти путь кодового дерева, вдоль которого цена пути возрастает. Поскольку при этом неправильные пути, вдоль которых цена пути в течение длительного промежутка времени постоянно возрастает, встречаются очень редко, то оказывается очень большой вероятность того, что путь кодового дерева, вдоль которого цена пути возрастает, определяет переданную последовательность.

6.2.2. Поведение декодера

Решение о том, возрастает или не возрастает цена пути, декодер принимает, проводя сравнение цены пути с порогом. При этом в каждом узле кодового дерева декодер упорядочивает y ребер, выходящих из этого узла, в порядке приращения цены пути вдоль этих ребер. Декодер может находиться в следующих состояниях:

I. «*Смотри вперед*» (СВ).

Находясь в этом состоянии, декодер считает, что путь $\bar{u}_n$, ведущий к узлу, в котором он находится в настоящий момент, является правильным, и принимает решение о переходе к пути $\bar{u}_{n+1}$, получающемуся присоединением к $\bar{u}_n$ следующего ребра [1]).

II. «*Смотри назад*» (СН).

Декодер считает, что правильным является префикс $\bar{u}_m$ из некоторых m первых ребер пути $\bar{u}_n$, и принимает решение продолжить поиск правильного пути среди других путей, являющихся продолжением $\bar{u}_m$.

Текущее значение порога в состоянии СВ выбирается непосредственно под текущим значением $\gamma(\bar{u}_n, \bar{v}_n)$. Предположим, что декодер достиг некоторого узла, которому соответствует путь u_n на кодовом дереве, и находится в состоянии СВ. В этом узле декодер выбирает первое по порядку ребро (как указывалось выше, ребра упорядочены), вычисляет приращение цены пути вдоль этого ребра и смотрит, превышает ли цена пути (пути, включающего присоединенное ребро) текущий порог. Если цена пути, включающего присоединенное ребро, превы-

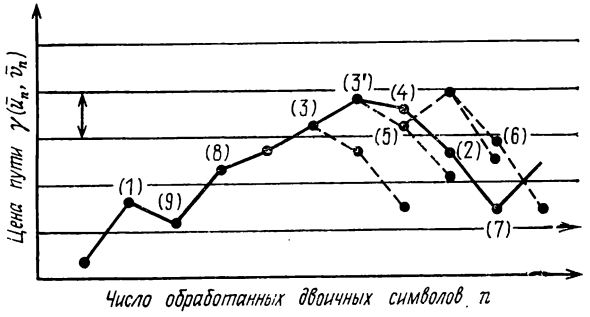
[1]) Здесь индекс n обозначает число ребер, составляющих путь, а не число символов, как в (6.1). — *Прим. ред.*

шает текущий порог, то проведенное обследование считается удачным и декодер переходит вдоль этого ребра в новый узел. При этом присоединенное ребро записывается в память декодера, а порог с шагом D_0 поднимается вверх до максимально возможной величины, меньшей, чем цена пути с присоединенным ребром. Далее, точно так же декодер выбирает одно из ребер, являющихся продолжением ребра, присоединенного на предыдущем шаге, сравнивает новую цену пути с порогом, и если эта новая цена пути превышает порог, то новое ребро считается правильным и декодер переходит к обследованию последующих ребер. До тех пор пока цены исследуемых путей не опускаются ниже соответствующих порогов, декодер движется вперед по дереву.

Предположим, что декодер в состоянии СВ обнаружил, что цена пути, вычисленная вдоль ребра, выходящего из некоторого $(n+1)$-го узла, лежит ниже текущего порога. В этом случае декодер принимает решение о том, что он оказался на неверном пути, и переходит в состояние СН. При этом декодер тем не менее считает правильным путь до «самого старого» узла m, цена пути в котором еще лежит выше текущего значения порога. Последнее значение порога фиксируется, и декодер пытается найти другой путь, проходящий через узел m, но лежащий выше фиксированного значения порога. Для этого декодер сначала возвращается из узла, в котором он перешел в состояние СН, на одно ребро назад и осуществляет поиск другого ребра, лежащего выше замороженного порога. Если такое ребро удается найти, то декодер возвращается в состояние СВ и начинает вновь двигаться вперед. Если указанное ребро обнаружить не удается, то декодер возвращается из $(n+1)$-го узла на два ребра назад и вновь пытается найти путь, лежащий выше фиксированного порога. При неблагоприятных условиях декодер может вернуться в узел m, так и не найдя пути, проходящего через узел m и лежащего выше фиксированного порога. В этом случае декодер вновь переходит в узел $n+1$, снижает значение порога на D_0, переходит в состояние СВ и пытается продвинуться вперед при пониженном пороге. Если все ребра, выходящие из узла $n+1$, пересекают это новое значение порога, то декодер повторяет указанные выше действия с этим новым фиксированным порогом. Будем считать, что, выполняя достаточно большое число раз указанные выше действия, декодер в конце концов найдет правильный путь.

Фиг. 6.7 иллюстрирует описанный выше алгоритм. Рассмотрим изменение величины $\gamma(\bar{u}_n, \bar{v}_n)$ при движении слева направо. Так как вначале величина $\gamma(\bar{u}_n, \bar{v}_n)$ возрастает, то декодер находится в состоянии СВ и считает пройденный путь правильным. В узле (*1*) значение $\gamma(\bar{u}_n, \bar{v}_n)$ несколько уменьшается, но тем не

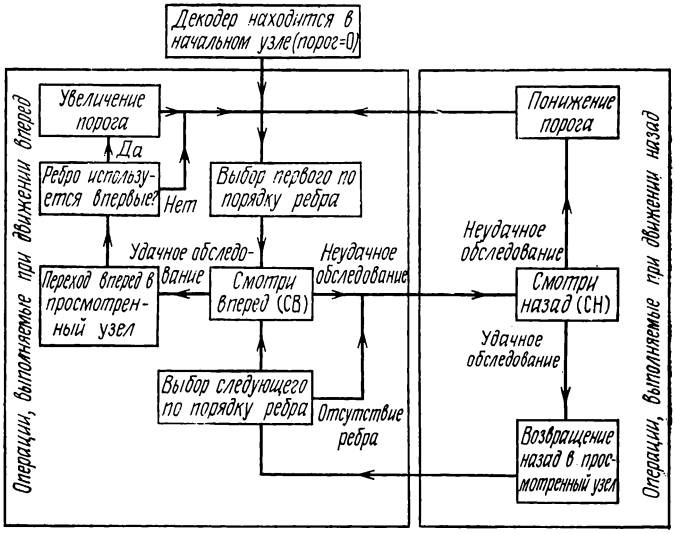
менее оказывается выше порога и декодер остается в состоянии СВ. Однако в узле (2) значение $\gamma(\bar{u}_n, \bar{v}_n)$ становится меньше текущего значения порога, равного $3D_0$. Декодер переходит в состояние СН, но путь до самого левого узла, лежащего выше этого порога [т. е. узла (3)] считает правильным. Вначале декодер возвращается из узла (2) в узел (4) и начинает поиск правильного пути из этого узла, но ни одно ребро, выходящее из узла (4), не лежит выше порога. Декодер возвращается в узел (3') и продолжает поиск из этого узла. Путь, оканчивающийся в узле (5), оказывается лежащим выше порога, и декодер переходит в состояние СВ. Однако в узле (6) цена пути вновь оказывается ниже порога, декодер возвращается в состояние СН, но путь до узла (3) по-прежнему считает правильным.



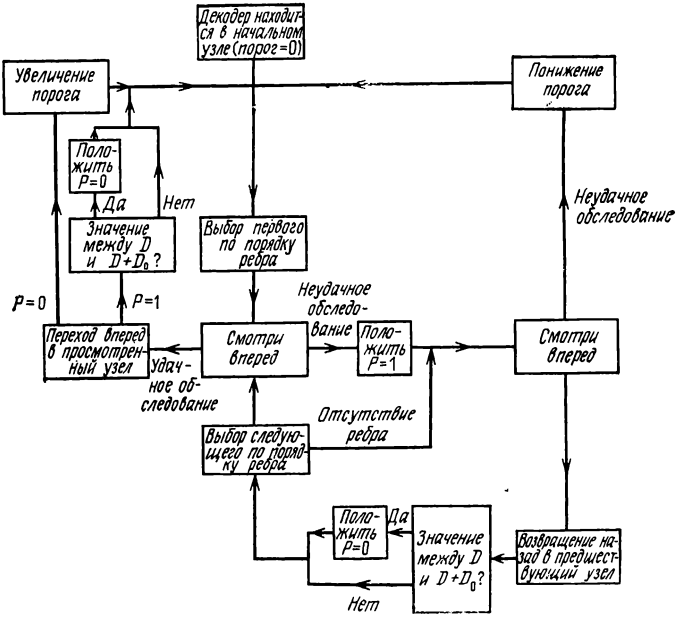
Фиг. 6.7. Пример к описанию алгоритма Фано.

После того как становится ясным, что все пути, выходящие из узла (3), пересекают порог $3D_0$, декодер возвращается в узел (2), понижает порог до величины $2D_0$ и возвращается в состояние СВ. Выполняя последние операции, декодер предполагает, что в окрестности узла (2) возникли сильные шумы, в результате чего правильный путь оказался похожим на неправильные пути. В этой ситуации декодер пытается найти путь, выходящий из узла (2) и лежащий выше порога $2D_0$. Если, выполняя процедуру, аналогичную описанной выше, декодер не находит узла, лежащего выше порога $2D_0$, то он понижает порог до величины D_0, переходит в состояние СВ и начинает движение вперед из узла (7).

На фиг. 6.8 показана логическая блок-схема алгоритма Фано. Смысл операций СВ, СН, «переход вперед в просмотренный узел», «возвращение назад в просмотренный узел», «увеличение порога», «понижение порога» должен быть понятен из приведенного выше описания алгоритма. Так как в каждом узле ребра упорядочены в соответствии с приращением вдоль них цены пути, то ясными оказываются и операции «выбор первого



Фиг. 6.8. Логическая схема алгоритма Фано.



Фиг. 6.9. Логическая схема алгоритма Фано (с функцией P).

по порядку ребра» и «выбор следующего по порядку ребра». Наличие вопроса «ребро используется впервые?» связано с необходимостью фиксировать порог в состоянии СН и предотвратить остановку алгоритма из-за зацикливания. Фано предложил оригинальный способ получения ответа на этот вопрос, который иллюстрируется на фиг 6.9.

Функция P определяется таким образом, что $P = 1$ в состоянии СН и $P = 0$ в остальных случаях. Если $P = 0$, то любой путь, не пересекающий текущее значение порога, исследуется впервые. В случае $P = 1$ декодер впервые переходит узел в следующих случаях:

1) После продвижения вперед цена пути оказывается лежащей между текущим значением порога D и величиной $D + D_0$, например в приведенном выше примере при переходе из узла (3′) в узел (5).

2) Декодер возвращается назад к некоторому пути, цена которого лежит между текущим значением порога и величиной $D + D_0$. Так, в приведенном выше примере при переходе в узел (7) цена пути оказывается лежащей ниже порога $2D_0$, декодер возвращается в узел (2) и начинает поиск другого пути. Не найдя пути, исходящего из узла (2), декодер продолжает поиск из узла (4). Если и в этом случае поиск также оказывается безуспешным, декодер начинает поиск из узла (3′). Путь из узла (3) в узел (5) уже исследовался раньше, и это легко обнаружить, поскольку цена пути в узле (5) лежит между значениями порога $3D_0$ и $4D_0$, а не $2D_0$ и $3D_0$.

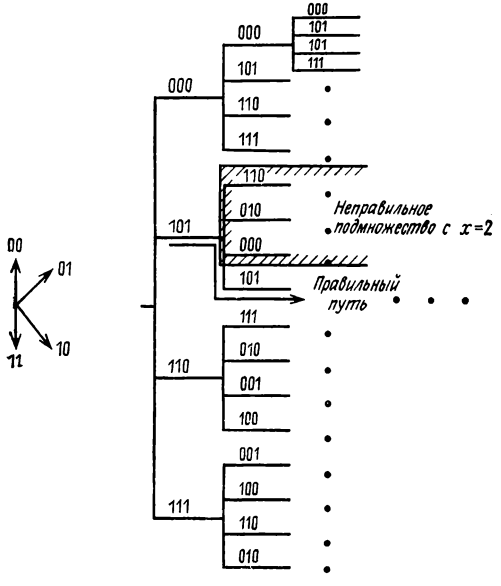
Путь с узлом (7) не может быть достигнут при значении порога $D + D_0$, и, следовательно, он является новым. При переходе в него значение P изменяется с 1 на 0.

6.3. Среднее число операций при декодировании

6.3.1. Среднее число операций

При последовательном декодировании с помощью алгоритма Фано среднее число операций при декодировании является случайной величиной, зависящей от принятой последовательности, подвергшейся воздействию шумов в канале, переданной информационной последовательности и сверточного кода. В силу этого определение среднего числа операций для некоторого заданного кода является очень трудной задачей. Однако с помощью границы случайного кодирования Сэваджу [9, 10] удалось получить границу для среднего по некоторому ансамблю кодов числа операций при декодировании.

Уточним сначала ряд терминов. Путь на кодовом дереве, соответствующий переданной информационной последовательности, будем называть правильным путем. Пути, начинающиеся в узлах правильного пути, но отличающиеся от правильного пути, будем называть неправильными путями. Совокупность неправильных путей, начинающихся в x-ом узле правильного пути, назовем x-ым подмножеством неправильных путей. На фиг. 6.10 показано 2-е подмножество неправильных путей.



Фиг. 6.10. Неправильное подмножество с $x = 2$.

Вывод границы для среднего числа операций условно можно разделить на две части. Сначала в предположении, что декодер достиг x-го узла правильного пути, находится верхняя граница для среднего числа операций, выполняемых декодером при движении из x-го узла по правильному пути и подмножеству неправильных путей. Далее с помощью границы случайного кодирования находится граница для среднего числа операций.

Каждый путь в x-м подмножестве неправильных путей однозначно определяется двумя числами: числом t, показывающим его длину в ребрах, и числом m, показывающим его положение среди путей той же длины. Если число ребер, выходящих из каждого узла кодового дерева, обозначить через y, то число путей длины t и менее $\tau(t)$, содержащихся в x-м подмножестве

неправильных путей, будет определяться равенством

$$\tau(t)=\begin{cases}(y-1)\,y^{t-1}, & t\geqslant 1,\\ 1 & t=0\end{cases} \tag{6.9}$$

(равенство $\tau(0)=1$ принято для удобства).

Ниже для простоты будем предполагать, что $x=1$. Пусть $U_{t,m}$ — путь, задаваемый парой целых чисел (t, m); U_w^* — правильный путь длины w ребер; V_t — принятая последовательность длины t ребер. Допустим, что текущее значение порога декодера равно jD_0. Тогда если $\gamma(U_{t,m}, V_t) < jD_0$, то декодер достичь пути (t, m) не может. Попытаемся найти среднее число операций, выполняемых декодером на пути (t, m) в случае, если $\gamma(U_{t,m}, V_t) \geqslant jD_0$.

1) Если декодер достиг пути (t, m) впервые, то он находится в состоянии СВ, и для того, чтобы пройти одно следующее ребро, он должен выполнить одну операцию.

2) Допустим, что двоичная переменная P алгоритма Фано равна 1 и декодер при поиске когда-то возвращается к пути (t, m). В силу того что $P=1$, декодер не может вторично выбрать ребро, указанное в п. 1, и пытается найти какое-либо другое ребро, чтобы пройти вперед. Если этот поиск оканчивается неудачно, то декодер возвращается в узел, предшествующий началу (t, m). Следовательно, при значении порога jD_0 и условии, что $\gamma(U_{t,m}, V_t) \geqslant jD_0$, на пути (t, m) декодер может выполнить самое большое $(y+1)$ операций.

Далее, для того чтобы получить оценку сверху для числа операций c, рассмотрим необходимые условия достижения пути (t, m). Пусть Γ — минимальная цена правильного пути, а именно

$$\Gamma=\min_{0\leqslant w}\gamma(U_w^*, V_w), \tag{6.10}$$

и пусть D_Γ — значение порога, лежащее непосредственно под ценой Γ. Необходимым условием достижения пути (t, m) является выполнение следующего неравенства:

$$\gamma(U_{t,m}, V_t)-(\Gamma+jD_0)\geqslant 0. \tag{6.11}$$

Действительно:

1) в случае $j\geqslant 1$ путь (t, m) может быть достигнут при значении порога jD_0;

2) если $j=0$, то $\gamma(U_{t,m}, V_t)\geqslant\Gamma\geqslant D_\Gamma$ и путь (t, m) может быть достигнут при значении порога D_Γ;

3) так как при $j=-1$ $\gamma(U_{t,m}, V_t)>\Gamma-D_0$ и $\Gamma-D_0\geqslant D_\Gamma$, то $\gamma(U_{t,m}, V_t)>\Gamma-D_0>D_\Gamma$ и путь (t, m) может быть достигнут при значении порога $\Gamma-D_0$.

Таким образом, путь (t, m) может быть достигнут только при значениях порога $\Gamma+jD_0$, $j\geqslant -1$.

Величины t и m, определяющие путь (t, m), могут принимать значения соответственно от 0 до ∞ и от 1 до $\tau(t)$. Таким образом, вводя ступенчатую функцию

$$W(x)=\begin{cases} 1, & x\geqslant 0, \\ 0, & x<0, \end{cases} \tag{6.12}$$

число операций c можно оценить сверху следующим образом:

$$c\leqslant(y+1)\sum_{j=-1}^{\infty}\sum_{t=0}^{\infty}\sum_{m=1}^{\tau(t)} W[\gamma(U_{t,m}, V_t)-\Gamma-jD_0]. \tag{6.13}$$

Далее оценим сверху правую часть (6.13). Величина Γ, согласно формуле (6.10), определяется равенством

$$\Gamma=\min_{0\leqslant w}\gamma(U_w^*, V_w), \tag{6.14}$$

т. е. Γ является значением $\gamma(U_w^*, V_w)$ при некотором $w=0, 1, \ldots, \infty$. Так как $W(x)$ является ступенчатой функцией, принимающей значения 0 и 1, то среди значений $W[\gamma(U_{t,m}, V_t)-\gamma(U_w^*, V_w)-jD_0]$, $w=0, 1, \ldots, \infty$, содержится и $W[\gamma(U_{t,m}, V_t)-\Gamma-jD_0]$. Таким образом, для c получаем следующую оценку:

$$c\leqslant(y+1)\sum_{j=-1}^{\infty}\sum_{w=1}^{\infty}\sum_{t=0}^{\infty}\sum_{m=1}^{\tau(t)} W[\gamma(U_{t,m}, V_t)-\gamma(U_w^*, V_w)-jD_0]. \tag{6.15}$$

Аналогичную оценку можно получить для произвольного $x\geqslant 1$, если несколько изменить определения величин $\gamma(U_{t,m}, V_t)$, $\gamma(U_w^*, V_w^*)$ и др. Действительно, пусть γ_x — значение правильного пути из x первых ребер, $\gamma(U_{t,m}, V_t)$ — значение пути из t последних ребер пути (t, m), входящего в x-е подмножество неправильных путей. Тогда $\gamma_x+\gamma(U_{t,m}, V_t)$ — полная цена этого пути. Пусть $\gamma(U_w^*, V_w)$ — цена пути из w последних ребер правильного пути длины $x-1+r$ ребер. Тогда $\gamma_x+\gamma(U_w^*, V_w)$ — полная цена правильного пути. Следовательно, число операций, выполняемых декодером на пути (t, m), входящем в x-е подмножество неправильных путей, зависит от цены $\gamma_x+\gamma(U_{t,m}, V_t)$ и минимальной цены правильного пути $\gamma_x+\Gamma$, где $\Gamma=\min_{w\geqslant 0}\gamma(U_w^*, V_w)$. Однако так как $\gamma_x+\gamma(U_{t,m}, V_t)-\gamma_x-\Gamma=\gamma(U_{t,m}, V_t)-\Gamma$, то, используя определенные вышеуказанным образом величины $\gamma(U_{t,m}, V_t)$ и $\gamma(U_w^*, V_w)$, можно доказать справедливость соотношения (6.13) и для x-го подмножества неправильных путей.

6.3.2. Свойство независимости в древовидном коде

На следующем этапе вывода верхней границы для среднего числа операций при декодировании используется метод случайного кодирования и проводится усреднение (6.15) по ансамблю кодов, шумам в канале и информационным последовательностям. Чтобы можно было применить метод случайного кодирования, потребуется ансамбль древовидных кодов, обладающий следующими свойствами:

I. Кодовые слова, или, точнее, символы ребер, принадлежащих различным путям кодового дерева, должны быть статистически независимыми. А именно для любых двух путей U_1 и U_2 должно выполняться равенство

$$Q(U_1, U_2) = Q(U_1)\, Q(U_2).$$

II. Символы канала каждого кодового слова должны быть статистически независимыми. А именно для любой кодовой последовательности $U_n = (u_1, u_2, \ldots, u_n)$ длины n должно выполняться равенство

$$Q(U_n) = Q(u_1) Q(u_2) \ldots Q(u_n).$$

Приведенная несколько ниже лемма, принадлежащая Галлагеру [11], дает метод построения ансамбля древовидных кодов, обладающих указанными выше свойствами.

Пусть a — некоторое целое число; рассмотрим двоичный сверточный кодер с k_0 входами и an_0 выходами. Пусть $i_0^{(1)}, \ldots, i_0^{(k_0)}$; $i_1^{(1)}, \ldots, i_1^{(k_0)}$; ... и $t_0^{(1)}, \ldots, t_0^{(an_0)}$; $t_1^{(1)}, \ldots, t_1^{(an_0)}$; ... — последовательности символов соответственно на входах и выходах этого кодера. Согласно формуле (5.14), входные и выходные символы сверточного кодера связаны между собой следующим соотношением:

$$t_u^{(i)} = \sum_{l=0}^{m} \sum_{j=1}^{k_0} g_{(i)}^{(j)}\, i_{u-l}^{(j)}. \tag{6.16}$$

В целях упрощения далее будем считать m равным бесконечности. Выходную последовательность кодера сложим с произвольной двоичной последовательностью $Z = z_0^{(1)}, \ldots, z_0^{(an_0)}; z_1^{(1)}, \ldots, z_1^{(an_0)}; \ldots$ и построим таким образом последовательность $t_0^{(1)} \oplus z_0^{(1)}, \ldots, t_0^{(an_0)} \oplus z_0^{(an_0)}; t_1^{(1)} \oplus z_1^{(1)}, \ldots, t_1^{(an_0)} \oplus z_1^{(an_0)}$. Эту последовательность разобьем на блоки по a двоичных символов, и каждый из полученных блоков будем передавать по каналу с помощью одного из его символов. Вероятность появления на входе канала k-го символа канала обозначим через q_k. В данном случае n_0 символов канала порождаются k_0 двоич-

ными информационными символами, и, следовательно, рассматриваемый код имеет скорость передачи $R = k_0/n_0$.

Предположим, что параметры k_0, n_0 и a, а также соответствие между двоичными последовательностями длины a и символами канала заданы. Рассмотрим ансамбль кодов, который получается в том случае, если все коэффициенты $g^{(j)}_{(i),\,t}$ в формуле (6.16) и все компоненты последовательности z являются независимыми двоичными случайными величинами, принимающими значения 0 и 1 с равными вероятностями.

Лемма 6.1. (Галлагер.) *В рассмотренном выше ансамбле кодов кодовая последовательность* $u_0^{(1)}, \ldots, u_0^{(k_0)}; u_1^{(1)}, \ldots, u_1^{(k_0)}; \ldots,$ *соответствующая произвольной, но заданной информационной последовательности, является случайной последовательностью, символы которой являются независимыми случайными величинами, принимающими значение* k *с вероятностью* q_k, $k = 1, \ldots, K$. *Если информационные последовательности* l *и* l' *совпадают лишь в первых* $b - 1$ *блоках, то соответствующие им кодовые последовательности также совпадают в первых* $b - 1$ *блоках, а символы остальных блоков этих кодовых последовательностей являются статистически независимыми.*

Доказательство. Так как последовательность Z является последовательностью независимых двоичных случайных величин, принимающих значения 0 и 1 с равными вероятностями, то при произвольной фиксированной последовательности на выходе двоичного сверточного кодера последовательность $T \oplus Z$ также является последовательностью независимых двоичных случайных величин, принимающих значения 0 и 1 с равными вероятностями. В силу этого кодовая последовательность на входе канала представляет собой последовательность независимых символов канала, k-й из которых имеет вероятность q_k, $k = 1, \ldots, K$.

Далее, пусть l и l' — информационные последовательности, символы которых начинают различаться лишь в b-м блоке. Если T и T' — соответствующие им последовательности на выходе двоичного сверточного кодера, то $T'' = T' \oplus T$ — выходная последовательность кодера, соответствующая входной последовательности $l'' = l' \oplus l$. Так как первые $b - 1$ блоков последовательности l'' нулевые, то нулевыми являются также и первые $b - 1$ блоков последовательности T''. Если только при некотором значении j, например j', имеет место равенство $i_b^{(j')} = 1$, то для произвольного $u \geqslant b$

$$i''^{(i)}_u = g^{(j')}_{(i),\,(u-b)} + \sum_{j \neq j'} g^{(j)}_{(i),\,(u-b)} i''^{(j)}_{(b)} + \sum_{l=b+1}^{u} \sum_{j=1}^{k_0} g^{(j)}_{(i),\,(u-l)} i''_l. \qquad (6.17)$$

Так как $g^{(j')}_{(i),\,(u-b)}$ — случайные величины, статистически независящие от других величин $g^{(j)}_{(i),\,l}$ и принимающие значения 0 и 1 с равными вероятностями, то символы $t''^{(i)}_u$ также оказываются двоичными случайными величинами, принимающими значения 0 и 1 с равными вероятностями. Более того, так как при $i \neq i'$ случайные величины $t''^{(i)}_u$ и $g^{(j')}_{(i),\,(u-b)}$ статистически независимы, то статистически независимыми являются величины $t''^{(i)}_u$ и $t''^{(i)}_s$ при всех $s < u$. Следовательно, все блоки последовательности T'', за исключением первых $b-1$, являются статистически независимыми, принимающими с равными вероятностями все возможные значения. Таким образом, символы последовательностей $T \oplus Z$ и $T' \oplus Z$, за исключением символов первых $b-1$ блоков, статистически независимы. Отсюда в свою очередь следует, что символы последовательностей T и T', за исключением символов первых $b-1$ блоков, также являются независимыми.

Выше в целях упрощения предполагалось, что кодовое ограничение имеет бесконечно большие значения. Поскольку кодовое ограничение сверточных кодов, порождаемых реальными кодерами, является конечным, то из-за этого, как описывается в следующем разделе, могут возникать необнаруживаемые ошибки. Однако при определении среднего числа операций предположение о бесконечности кодового ограничения позволяет упростить анализ, проводимый методом случайного кодирования, и, как будет показано ниже, получить для алгоритма Фано верхнюю границу для среднего числа операций, не зависящую от длины кодового ограничения n_A. Это показывает целесообразность предположения о бесконечности кодового ограничения при анализе среднего числа операций.

6.3.3. Верхняя граница для среднего числа операций

В этом разделе оценку среднего числа операций, задаваемую формулой (6.15), усредним по ансамблю кодов, обладающих указанными выше свойствами. Для этого нам понадобится следующее неравенство Чернова, которое позволяет оценить сверху ступенчатую функцию $W(x)$ с помощью экспоненциальной функции (фиг. 6.11)

$$W(x) \leqslant e^{\lambda x}, \quad \lambda \geqslant 0. \tag{6.18}$$

Здесь докажем следующую теорему.

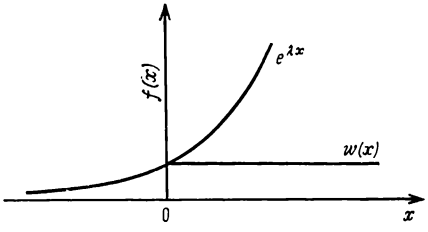
Теорема 6.1. (Сэвадж.) *В дискретном канале без памяти при использовании алгоритма Фано среднее по ансамблю кодов число операций при декодировании (приходящееся на одно де-*

кодированное ребро) ограничено сверху константой, не зависящей от длины кодового ограничения, если только $U < R_{\text{выч}}$ *и* $R < U/2 + R_{\text{выч}}/2$, *где*

$$R_{\text{выч}} = \max_{\{q_k\}} \left[-\ln \sum_{j=1}^{J} \left(\sum_{k=1}^{K} q_k \left[p\left(v_j \mid v_k\right) \right]^{1/2} \right)^2 \right]. \tag{6.19}$$

Доказательство. Подставим (6.18) в (6.15). Тогда

$$c \leqslant (y+1) \sum_{j=-1}^{\infty} \sum_{w=0}^{\infty} \sum_{t=0}^{\infty} \sum_{m=1}^{\tau(t)} e^{\lambda \left[\gamma\left(U_{t,\,m},\, V_t\right) - \gamma\left(U_w^*,\, V_w\right) - jD_0 \right]}. \tag{6.20}$$



Фиг. 6.11. $e^{\lambda x}$ и $w(x)$.

Так как каждому ребру кодового дерева соответствует n_0 символов канала, то в данном случае

$$\gamma\left(U_{t,\,m},\, V_t\right) = I\left(U_{t,\,m},\, V_t\right) - tn_0U, \tag{6.21}$$

$$\gamma\left(U_w^*,\, V_w\right) = I\left(U_w^*,\, V_w\right) - wn_0U. \tag{6.22}$$

Из формул (6.20) — (6.22) имеем

$$c \leqslant (y+1) \sum_{j=-1}^{\infty} \sum_{w=0}^{\infty} \sum_{t=0}^{\infty} \sum_{m=1}^{\tau(t)} e^{-j\lambda D_0} e^{-tn_0U\lambda} e^{wn_0U\lambda} e^{\lambda I\left(U_{t,\,m},\, V_t\right)} e^{-\lambda I\left(U_w^*,\, V_w\right)},$$

где

$$e^{\lambda I\left(U_{t,\,m},\, V_t\right)} = e^{(\lambda/\ln 2)\ln I\left(U_{t,\,m},\, V_t\right)} = \left[\frac{p\left(V_{t,\,m} \mid U_{t,\,m}\right)}{Q\left(V_t\right)} \right]^{\lambda/\ln 2},$$

$$e^{-\lambda I\left(U_w^*,\, V_w\right)} = e^{(-\lambda/\ln 2)\ln I\left(U_w^*,\, V_w\right)} = \left[\frac{Q\left(V_w\right)}{p\left(V_w \mid U_w\right)} \right]^{\lambda/\ln 2}.$$

Суммируя по j в правой части последнего неравенства, получаем

$$c \leqslant (y+1) B \sum_{w=0}^{\infty} \sum_{t=0}^{\infty} \sum_{m=1}^{\tau(t)} e^{-tn_0U\nu \ln 2} e^{wn_0U\nu \ln 2} \times$$

$$\times \left[\frac{p\left(V_t \mid U_{t,\,m}\right)}{Q\left(V_t\right)} \right]^{\nu} \left[\frac{Q\left(V_w\right)}{p\left(V_w \mid U_w^*\right)} \right]^{\nu}, \tag{6.23}$$

где $\nu = \lambda/\ln 2$ и $B = e^{\lambda D_0}/(1 - e^{-\lambda D_0})$.

Далее зафиксируем правильный путь U_t и усредним (6.23) по принимаемым последовательностям V_w. Так как среднее значение суммы равно сумме средних значений и, кроме того, V_t входит только в два последних сомножителя правой части последнего неравенства, то усреднение (6.23) сводится к нахождению среднего значения $[p]_{\text{ср}}$ произведения $[p(V_t \mid U_{t,m})/Q(V_t)]^{\nu}[Q(V_w)/p(V_w \mid U_w^*)]^{\nu}$. Имеет место следующее равенство:

$$[p]_{\text{ср}} = \overline{\left[\frac{p(V_t \mid U_{t,m})}{Q(V_t)}\right]^{\nu}\left[\frac{Q(V_w)}{p(U_w \mid U_w^*)}\right]^{\nu}} = \sum_{V_t} p(V_t \mid U_t)\left[\frac{p(V_t \mid U_{t,m})}{Q(V_t)}\right]^{\nu}\left[\frac{Q(V_w)}{p(V_w \mid U_w^*)}\right]^{\nu}. \quad (6.24)$$

Так как канал является дискретным каналом без памяти, то при всех $t \geqslant r$

$$[p]_{\text{ср}} = \sum_{V_t} p(V_{t-w} \mid U_{t-w}^*)\, p(V_w \mid U_w^*)\left[\frac{p(V_t \mid U_{t,m})}{Q(V_t)}\right]^{\nu}\left[\frac{Q(V_w)}{p(V_w \mid U_w^*)}\right]^{\nu} = \sum_{V_t} \frac{p(V_{t-w} \mid U_{t-w}^*)}{[Q(V_{t-w})]^{\nu}}[p(V_w \mid U_w^*)]^{1-\nu}[p(V_t \mid U_{t,m})]^{\nu}, \quad (6.25)$$

где через U_{t-w}^* и V_{t-w} обозначены соответственно последние $t-w$ ребер путей U_t^* и V_t.

Далее проведем усреднение (6.25) по ансамблю кодов. Если воспользоваться свойствами рассматриваемого ансамбля кодов, даваемыми леммой 6.1, а именно тем, что различные пути кодового дерева статистически независимы и, кроме того, независимо с распределением вероятностей $\{q_k, 1 \leqslant k \leqslant K\}$ выбираются и символы канала для каждого фиксированного пути, то усреднение (6.25) по U_{t-w}^*, U_w^* и $U_{t,m}$ можно провести раздельно. Обозначим искомое среднее значение через $\overline{[p]}_{\text{ср}}$. В силу вышеизложенного

$$\overline{[p]}_{\text{ср}} = \sum_{V_t} \frac{1}{[Q(V_{t-w})]^{\nu}}\overline{p(V_{t-w} \mid U_{t-w}^*)}\,\overline{[p(V_w \mid U_w^*)]^{1-\nu}}\,\overline{[p(V_t \mid U_{tm})]^{\nu}}, \quad (6.26)$$

где

$$\overline{p(V_{t-w} \mid U_{t-w}^*)} = \sum_{U_{t-w}^*} Q(U_{t-w}^*)\, p(V_{t-w} \mid U_{t-w}^*) = Q(V_{t-w}) \leqslant 1. \quad (6.27)$$

Следовательно,

$$\overline{[p]}_{\text{ср}} \leqslant \sum_{V_t} \overline{[p(V_w \mid U_w^*)]^{1-\nu}}\, \overline{[p(V_t \mid U_{t-w})]^{\nu}}. \tag{6.28}$$

Правую часть (6.28) оценим сверху, воспользовавшись неравенством Шварца:

$$\sum_{V_t} \overline{[p(V_w \mid U_w^*)]^{1-\nu}}\, \overline{[p(V_t \mid U_{t,m})]^{\nu}} \leqslant$$
$$\leqslant \left[\sum_{V_w} \overline{\{p(V_w \mid U_w^*)^{1-\nu}\}^2}\right]^{1/2} \left[\sum_{V_t} \overline{\{p(V_t \mid U_{t,m})^{\nu}\}^2}\right]^{1/2}. \tag{6.29}$$

Так как в случае дискретного канала без памяти вероятности $p(V_w \mid U_w^*)$ и $p(V_t \mid U_{t,m})$ являются произведениями соответствующих вероятностей для пар символов u_i и v_t из U и V, то

$$\left[\sum_{V_w} \overline{\{p(V_w \mid U_w^*)^{1-\nu}\}^2}\right]^{1/2} = \left[\sum_{j=1}^{J} \left\{\sum_{k=1}^{K} q_k p(v_j \mid u_k)^{1-\nu}\right\}^2\right]^{wn_0/2},$$
$$\left[\sum_{V_w} \overline{\{p(V_t \mid U_{t,m})^{\nu}\}^2}\right]^{1/2} = \left[\sum_{j=1}^{J} \left\{\sum_{k=1}^{K} q_k p(v_j \mid u_k)^{\nu}\right\}^2\right]^{tn_0/2}. \tag{6.30}$$

Выше был рассмотрен случай, когда $t \geqslant r$. В действительности аналогичные оценки имеют место и при $w \geqslant t$. В силу этого $\overline{[p]}_{\text{ср}}$, а следовательно, и $\bar{e}$ можно оценить сверху с помощью правых частей (6.30).

Чтобы получить по возможности более точную границу, правые части (6.30) необходимо минимизировать по ν и распределению вероятностей $\{q_k, 1 \leqslant k \leqslant K\}$. Однако здесь минимизацию по ν мы проводить не будем, положим просто $\nu = 1/2$ и минимизируем правые части (6.30) только по q_k. Пусть

$$R_{\text{выч}} = \max_{\{q_k\}} \left[-\ln \sum_{j=1}^{J} \left(q_k [p(v_j \mid u_k)]^{1/2}\right)^2\right]. \tag{6.31}$$

Тогда

$$\max_{\{q_k\}} \left[\sum_{j=1}^{J} \left(\sum_{k=1}^{K} q_k [p(v_j \mid u_k)]^{1/2}\right)^2\right]^{-1} =$$
$$= e^{\max\limits_{\{q_k\}} \left[-\log_2 \sum\limits_{j=1}^{J} \left(\sum\limits_{k=1}^{K} q_k [p(v_j \mid u_k)]^{1/2}\right)^2\right]} = e^{R_{\text{выч}}}.$$

Отсюда и из неравенств

$$\tau(t) = (y-1)\, y^{t-1} \leqslant y^t \leqslant e^{(n_0 \ln 2) R} \tag{6.32}$$

получаем

$$\bar{c} \leqslant (y+1) B \left(\sum_{t=0}^{\infty} \tau(t)\, e^{-tn_0 U (\ln 2/2)} e^{tn_0 (\ln 2/2)(-R_{\text{выч}})} \right) \times$$

$$\times \left(\sum_{w=0}^{\infty} e^{wn_0 U \ln 2/2} e^{-wn_0 \ln 2/2 R_{\text{выч}}} \right) \leqslant$$

$$\leqslant (y+1) B \left\{ 1 - e^{-(n_0 \ln 2/2)(R_{\text{выч}} + U - 2R)} \right\}^{-1} \left\{ 1 - e^{-(n_0 \ln 2/2)(R_{\text{выч}} - U)} \right\} \quad (6.33)$$

Из этого неравенства, в частности, следует справедливость теоремы 6.1.

6.4. Распределение числа операций и вероятность переполнения буфера

Как будет показано ниже, при последовательном декодировании в некоторой области скоростей передачи вероятность ошибки стремится экспоненциально к нулю с ростом длины кодового ограничения. Однако вероятность ошибки при декодировании не является тем параметром, который определяет поведение алгоритмов последовательного декодирования. Чтобы найти правильный путь при наличии сильных шумов, приемник должен обследовать большое число ребер, а для этого он должен хранить в памяти достаточно большое число принятых символов. Емкость памяти декодера в реальных системах ограничена, и поэтому при возникновении пачки шумов все элементы памяти могут оказываться занятыми, т. е. возникает так называемое переполнение буфера. Вероятность переполнения буфера является обычно тем параметром, который определяет поведение алгоритмов последовательного декодирования.

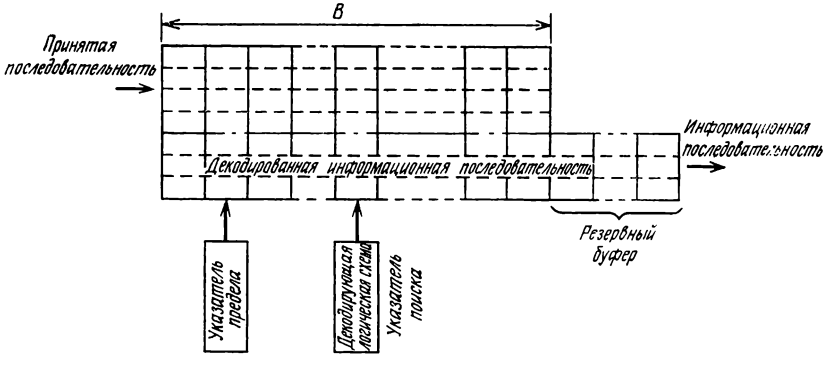
В предыдущем разделе было найдено среднее число операций, необходимых для декодирования одного ребра, однако величиной, определяющей переполнение буфера, является не среднее число операций, а распределение числа операций, а именно вероятность $p[c \geqslant L]$ того, что число операций при декодировании c превысит значение L. Как уже было показано выше, случайная величина c, определяющая число операций, выполняемых декодером в неправильных подмножествах кодового дерева, связана с ценами путей всех возможных длин; поскольку эти пути являются зависимыми, то нахождение точных значений вероятности $p[c \geqslant L]$ является трудной задачей. Поэтому очень важно попытаться обойти статистическую зависимость неправильных путей кодового дерева и получить верхнюю и нижнюю границы для вероятности $p[c \geqslant L]$, которые бы достаточно точно описывали ее поведение. Ниже такие границы будут получены, и из этих границ будет следовать, что распределение числа

операций при декодировании является распределением Парето, а именно $p[c \geqslant L] \approx L^{-\nu}$.

Исследование вероятности переполнения буфера проведем в три этапа. Сначала рассмотрим структуру буфера и его функционирование, затем получим верхнюю и нижнюю границы для вероятности $p[c \geqslant L]$ и наконец установим связь между вероятностью переполнения буфера и вероятностью $p[c \geqslant L]$.

6.4.1. Структура и функционирование буфера

Структура буфера показана на фиг. 6.12. Как видно из этой фигуры, часть емкости буфера используется для хранения символов принимаемых последовательностей, которые вводятся



Фиг. 6.12. Стандартная структура буфера.

в буфер слева блоками в порядке поступления последних (каждый блок соответствует определенному ребру кодового дерева), а другая часть емкости буфера используется для хранения результатов предварительного декодирования принятых блоков.

Буфер содержит два указателя Один из них, так называемый указатель предела, указывает первое слева ребро, обследованное декодером. Другой указатель, так называемый указатель поиска, указывает положение, в котором в данный момент времени декодер осуществляет поиск. Согласно фиг. 6.12, декодер может хранить B ребер. Защитная память, показанная на фиг. 6.12 справа, устанавливается для того, чтобы при переполнении буфера можно было выделить ненадежно декодированные данные.

При слабых шумах оба указателя находятся, как правило, вблизи левого конца буфера, а также близко друг к другу. По мере возрастания уровня шумов, действующих в канале, оба указателя обычно смещаются вправо. При этом указатель по-

иска всегда располагается справа от указателя предела и расстояние между этими указателями возрастает медленнее, чем расстояние между указателем предела и левым концом буфера. Так как при малом уровне шумов цены правильного и неправильного путей сильно различаются, то в результате проводимого поиска сразу удается разделить правильный и неправильный пути. При пачке шумов цены правильного пути начинают падать и декодер, исследуя как падающий участок правильного пути, так и неправильные пути, цены которых лежат выше минимальной цены правильного пути, постепенно понижает порог. С уменьшением уровня шумов цены правильного пути начинают возрастать и декодер движется вперед, вновь повышая значения порога. Однако, возвращаясь к фиг. 6.7, заметим, что до этого декодер должен обследовать неправильные пути между узлами (7) и (9). Так как число ребер неправильных путей, лежащих в этой области, растет экспоненциально с ростом длины этого участка, то резко возрастает и объем проводимых вычислений. В результате указатель предела перемещается вправо на значительное расстояние от левого конца буфера, однако при этом расстояние между указателями предела и поиска оказывается несколько меньшим.

6.4.2. Нижняя граница для p[c ⩾ L]

Распределение $p[c \geqslant L]$ числа операций при декодировании Z двоичных информационных символов было найдено Джекобсом и Берлекэмпом [12]:

$$p[c \geqslant L] \approx L^{-\nu}. \tag{6.34}$$

Точнее, ими была получена нижняя граница для распределения $p[c \geqslant L]$ числа операций c, необходимых декодеру для декодирования Z первых информационных символов, из которой следовало приведенное выше приближенное равенство. Эта нижняя граница была получена методом, в основных чертах совпадающим с методом получения нижней границы для вероятности ошибки при декодировании списком, использованным Шенноном, Галлагером и Берлекэмпом [5].

Пусть задан дискретный канал без памяти, и пусть

$$E_0(\nu) = \max_{\{q_k\}} \left\{ -\ln \sum_{j=1}^{J} \left(\sum_{k=1}^{K} q_k p(v_j \mid u_k)^{1/1+\nu} \right)^{1+\nu} \right\}. \tag{6.35}$$

Верхнюю огибающую $E_0(\nu)$ будем обозначать через $\hat{E}_0(\nu)$. Для большого числа реальных каналов $\hat{E}_0(\nu) = E_0(\nu)$ и вместо $E_0(\nu)$ всегда можно использовать $\hat{E}_0(\nu)$. Использование $\hat{E}_0(\nu)$ в подобных случаях позволяет упростить нижнюю границу.

Рассмотрим передачу по дискретному каналу без памяти M равновероятных сообщений с помощью блокового кода из M кодовых слов, каждое из которых состоит из N входных символов канала. Приемник обрабатывает последовательности длины N, поступающие с выхода канала, и сопоставляет каждой из них список из L сообщений. Вероятность того, что в этом списке сообщений отсутствует действительно передававшееся сообщение, обозначается через $P_e(N, M, L)$. Пусть ν — произвольное целое число. Если число кодовых слов M и число сообщений L, включаемых при декодировании в список, удовлетворяют неравенству

$$\frac{M}{L} \geqslant \exp\{N\hat{E}_0'(\nu) + O_1(\sqrt{N})\}, \tag{6.36}$$

то, согласно формулам (1.68) и (1.69),

$$P_e(N, M, L) \geqslant \exp[-N\{\hat{E}_0(\nu) - \nu\hat{E}_0'(\nu)\} - O_2(\sqrt{N})], \tag{6.37}$$

где $\hat{E}_0'(\nu) = d\hat{E}_0/d\nu$, а $O_1(\sqrt{N})$ и $O_2(\sqrt{N})$ — функции, стремящиеся к нулю с ростом $\sqrt{N}$. Эта граница справедлива для произвольного блокового кода длины N и произвольного декодера со списком объема L.

Прежде чем применить эту границу для нахождения распределения числа операций при декодировании, введем следующие величины:

$$R = \ln M/N \text{ (т. е. } M = e^{NR}), \tag{6.38}$$

$$R_\nu = \hat{E}_0(\nu)/\nu. \tag{6.39}$$

Покажем, что вероятность $P_e(N, M, L)$ можно оценить сверху функцией, которая имеет тот же вид, что и распределение Парето.

Для этого рассмотрим блоковый код со скоростью $R > R_\nu$, зафиксируем L и выберем некоторым специальным указанным ниже образом N так, чтобы выполнялось неравенство (6.36). Сначала запишем неравенство (6.36) в следующем виде:

$$L \leqslant M\exp\{-N\hat{E}_0'(\nu) - O_1(\sqrt{N})\}. \tag{6.40}$$

Далее, подставим e^{NR_ν} в правую часть (6.40) вместо $M = e^{NR} (\geqslant e^{NR_\nu})$ и выберем минимально возможное N, такое, что

$$L \leqslant \exp[N\{R_\nu - \hat{E}_0'(\nu)\} - O_1(\sqrt{N})]. \tag{6.41}$$

Пусть X — некоторая постоянная и N — минимальное кратное X, для которого выполняется неравенство (6.41). При таком выборе N наряду с (6.41) имеет место и другое неравенство:

$$L \geqslant \exp[(N - X)\{R_\nu - \hat{E}_0'(\nu)\} - O_1(\sqrt{N - X}]. \tag{6.42}$$

При этом справедливы также неравенства (6.36) и (6.37). Подставляя выражение (6.39) в (6.37), получаем

$$P_e(N, M, L) \geqslant \exp\left[-N\nu\{R_\nu - \hat{E}_0'(\nu)\} - O_2(\sqrt{N})\right]. \quad (6.43)$$

Возьмем натуральный логарифм от обеих частей (6.42):

$$\ln L + O_1(\sqrt{N-X}) \geqslant (N-X)\{R_\nu - \hat{E}_0'(\nu)\}. \quad (6.44)$$

Используя эту формулу, преобразуем неравенство (6.43) к виду

$$P_e(N, M, L) \geqslant \exp\left[\frac{-N\nu}{N-X}\{\ln L + O_1(\sqrt{N-X})\} - O_2(\sqrt{N})\right] =$$

$$= \exp\left[\ln L^{\frac{-N\nu}{N-X}+\nu-\nu} - \frac{N\nu}{N-X}O_1(\sqrt{N-X}) - O_2(\sqrt{N})\right] =$$

$$= L^{-\nu}\exp\left[\frac{-\nu X}{N-X}\ln L - \frac{N\nu}{N-X}O_1(\sqrt{N-X}) - O_2(\sqrt{N})\right]. \quad (6.45)$$

Как следует из формулы (6.44), N имеет порядок $\ln L$:

$$N \sim \frac{\ln L}{R_\nu - \hat{E}_0'(\nu)} + O_3(\sqrt{\ln L}). \quad (6.46)$$

Используя это соотношение, запишем формулу (6.45) в виде

$$P_e(N, M, L) \geqslant L^{-\nu}\exp\{-O_4(\sqrt{\ln L})\}. \quad (6.47)$$

Эта оценка может быть использована следующим образом.

Предположим, что в начале принятой последовательности возникла пачка ошибок длины N. Допустим, что длина этой пачки является целым числом, кратным некоторой постоянной X и удовлетворяющим неравенствам (6.41) и (6.42). Обозначим через $\{U_{Ni}\}$ совокупность передаваемых последовательностей длины N, выходящих из начального узла кодового дерева. Эту совокупность, содержащую $M = e^{NR}$ последовательностей, можно рассматривать как блоковый код длины N со скоростью R. Обозначим через V_N первые N символов принятой последовательности. Для каждой фиксированной последовательности V_N совокупность M вероятностей $p(V_N|U_N)$ упорядочим в порядке их убывания. Пусть $I_B(V_N)$—совокупность индексов $M-L$ последних вероятностей из этого списка. Если принята последовательность V_N, то $I_B(V_N)$ указывает $M-L$ наименее правдоподобных сообщений или, другими словами, $I_B(V_N)$ выделяет совокупность сообщений, которые при декодировании списком по максимуму правдоподобия не включаются в список.

Условная вероятность того, что при передаче последовательности U_{Ni}, индекс i которой принадлежит $I_B(V_N)$, будет принята последовательность V_N, равна

$$\sum_{i \in I_B(V_N)} \frac{1}{M} p[V_N | U_{Ni}], \tag{6.48}$$

и, следовательно, при декодировании списком по максимуму правдоподобия вероятность ошибки определяется равенством

$$P_e(N, M, L) = \sum_{V_N} \sum_{i \in I_B(V_N)} \frac{1}{M} p[V_N | U_{Ni}]. \tag{6.49}$$

Неравенство (6.47) представляет собой нижнюю границу для этой вероятности.

Далее, предположим, что принята последовательность V_N, и рассмотрим, каким образом последовательный декодер осуществляет поиск правильного пути, выходящего из начального узла кодового дерева. Поведение любого последовательного декодера характеризуется следующими особенностями:

1) Поиск ребер осуществляется последовательно. Если в некотором узле кодового дерева декодер выберет ребро, которое раньше не обследовалось, то это решение не зависит от принятой последовательности, соответствующей ребрам, лежащим на кодовом дереве глубже.

2) Декодер выполняет по крайней мере одну операцию на каждом обследуемом пути.

В результате этого поиска передаваемые последовательности $\{U_{Ni}\}$ обследуются до полной длины N, и декодер после N-го символа продолжает двигаться вперед; здесь мы рассмотрим лишь пути кодового дерева длины N.

На время допустим, что на месте $(N+1)$-го символа имеется некий учитель, который указывает декодеру, выбрал ли он правильную последовательность U_{Ni} или некоторую неправильную последовательность. Если последовательность U_{Ni} является неправильным путем, то декодер установленным для данного алгоритма порядком возвращается назад и начинает обследование другого пути U_{Nj} до длины N. Если и этот путь является неправильным, то учитель вторично указывает декодеру на ошибку. Декодер вновь возвращается назад в исходное состояние и начинает обследование третьей последовательности и т. д. Другими словами, эти действия декодера для каждой фиксированной принятой последовательности V_N длины N можно интерпретировать как один из способов упорядочения последовательностей из $\{U_{Ni}\}$.

Результат упорядочения зависит от принятой последовательности V_N, передаваемых последовательностей U_{Ni} и алгоритма

последовательного декодирования. Однако результирующий порядок, согласно особенности 1, не зависит от символов передаваемой и принятой последовательностей, находящихся на расстоянии N и более от начального узла кодового дерева. Пусть $I_s(V_N)$ — множество индексов i последних $M-L$ последовательностей из $\{U_{Ni}\}$, упорядоченных указанным выше образом.

Согласно особенности 2 последовательного декодирования, в любом узле кодового дерева, который обследуется в процессе декодирования, декодер должен выполнить по крайней мере одну операцию. Следовательно, в случае, если передается последовательность, индекс которой принадлежит $I_s(V_N)$, а принята последовательность V_N, то декодер с учителем на глубине N кодового дерева должен выполнить по крайней мере L операций.

Таким образом, вероятность $Q_N(L)$ того, что декодер с учителем выполнит по крайней мере L операций при обследовании кодового дерева до глубины N, должна удовлетворять неравенству

$$Q_N(L) \geqslant \sum_{V_N} \sum_{i \in I_s(V_N)} \frac{1}{M} p(V_N \mid U_{Ni}). \tag{6.50}$$

Для каждой фиксированной последовательности V_N наборы $I_B(V_N)$ и $I_s(V_N)$ задают подмножества из $M-L$ кодовых слов множества $\{U_{Ni}\}$. Так как $I_B(V_N)$ определяет подмножество наименее правдоподобных кодовых слов, то

$$\sum_{i \in I_s(V_N)} p[V_N \mid U_{Ni}] \geqslant \sum_{i \in I_B(V_N)} p[V_N \mid U_{Ni}]$$

и, следовательно,

$$Q_N(L) \geqslant \sum_{V_N} \sum_{i \in I_N(V_N)} \frac{1}{M} p[V_N \mid U_{Ni}] \geqslant p_e(L) = \\ = L^{-\nu} \exp\{-O_4(\sqrt{\ln L})\}. \tag{6.51}$$

Эта оценка показывает, что распределение числа операций при декодировании является распределением Парето. Заметим, что полученная оценка является оценкой снизу числа операций, выполняемых декодером в узлах кодового дерева, лежащих на глубине N и менее. Так как в число этих операций входят также и операции, выполняемые декодером в подмножестве неправильных путей, то левая часть (6 51) также является оценкой снизу для распределения числа операций c, выполняемых в подмножестве неправильных путей, г. е.

$$p[c \geqslant L] \geqslant O_N(L) \geqslant L^{-\nu} \exp\{-O_4 \sqrt{\ln L}\}. \tag{6.52}$$

Таким образом, нижняя оценка для $p[c \geqslant L]$ получена.

6.4.3. Верхняя граница для p[c ⩾ L]

Для вывода верхней границы для $p[c \geqslant L]$ потребуются следующие два неравенства:

1) Неравенство Чебышева.

Пусть W — произвольная случайная величина, $\overline{W}$ — ее среднее значение и σ^2 — дисперсия. Тогда

$$p[|W - \overline{W}| \geqslant \varepsilon] \leqslant \frac{\sigma^2}{\varepsilon^2}. \tag{6.53}$$

Справедливо также следующее неравенство:

$$p[t \geqslant \delta] \leqslant \frac{\overline{t^\nu}}{\delta^\nu}, \tag{6.54}$$

где t — случайная величина, принимающая только отрицательные значения.

2) Неравенство Минковского.

Пусть $a_1, \ldots, a_K$ — неотрицательные случайные величины, принимающие с вероятностью Q_j соответственно значения $a_{j1}, \ldots, a_{jK}$. Тогда для любого $r > 1$

$$\left[\sum_j Q_j \left(\sum_k a_{jk}\right)^{1/r}\right]^r \geqslant \sum_k \left(\sum_j Q_j a_{jk}^{1/r}\right)^r \tag{6.55}$$

и, кроме того,

$$\left[\overline{\left(\sum_k a_k\right)^r}\right]^{1/r} \leqslant \sum_k \left[\overline{a_k^r}\right]^{1/r}. \tag{6.56}$$

Используя эти два неравенства и верхнюю границу для числа операций при декодировании, даваемую соотношением (6.15), Сэвадж [9] получил следующую верхнюю границу для $p[c \geqslant L]$.

Теорема 6.2. (Сэвадж). *Существует сверточный код со скоростью передачи R, для которого распределение $p[c \geqslant L]$ числа операций с для типичного подмножества неправильных путей в дискретном канале без памяти удовлетворяет неравенству*

$$p[c \geqslant L] \leqslant GL^{-R_{\text{выч}}/R}. \tag{6.57}$$

Доказательство. Рассматривая суммы по j, w, t и m в (6.15), т. е. в неравенстве

$$c \leqslant (y+1) \sum_{j=-1}^{\infty} \sum_{w=0}^{\infty} \sum_{t=0}^{\infty} \sum_{m=1}^{\tau(t)} W[\gamma(U_{t,m}, V_t) - \gamma(U_w^*, V_w) - jD_0],$$

как суммы по k неравенства (6.56), получаем

$$\overline{[c^{\mu}]}^{1/\mu} \leqslant (y+1) \sum_{j=-1}^{\infty} \sum_{w=0}^{\infty} \sum_{t=0}^{\infty} \sum_{m=1}^{\tau(t)} \overline{\{[W[\gamma(U_{t,m}, V_t) - \gamma(U_w^*, V_w) - jD_0]^{\mu}\}}^{1/\mu}, \quad (6.58)$$

где черта сверху означает усреднение по принимаемым последовательностям и ансамблю кодов. Здесь воспользуемся ансамблем кодов, в котором все символы кодовых слов являются независимыми величинами с распределением $\{q_K, 1 \leqslant k \leqslant K\}$. Аналогично тому, как было получено неравенство (6.33), можно показать, что

$$\overline{[c^{\mu}]}^{1/\mu} \leqslant (y+1) B \{1 - e^{-(n_0 \ln 2/2\mu)(R_{\text{выч}} - U)}\}^{-1} \times \times \{1 - e^{-(n_0 \ln 2/2\mu)(R_{\text{выч}} + U - 2R\mu)}\}^{-1}, \quad (6.59)$$

где

$$B = \frac{e^{D_0/2}}{1 - e^{D_0/2}}. \quad (6.60)$$

При $U < R_{\text{выч}}$ и $R < (U + R_{\text{выч}})/2\mu$ величина $\overline{[c^{\mu}]}^{1/\mu}$ и, следовательно, $\overline{[c^{\mu}]}$ сходятся. Это означает, что существует код, для которого момент μ-го порядка величины c не превышает некоторую константу G. Из неравенств $U < R_{\text{выч}}$ и $R < (U + R_{\text{выч}})/2\mu$ получаем, что $\mu < R_{\text{выч}}/R$. Согласно неравенству Чебышева, типичное для ансамбля кодов распределение $p[c \geqslant L]$ величины c должно удовлетворять неравенству

$$\overline{p[c \geqslant L]} \leqslant GL^{-R_{\text{выч}}/R}. \quad (6.61)$$

Это доказывает существование по крайней мере одного кода, распределение величины c для которого удовлетворяет (6.57).

Таким образом, мы получили верхнюю и нижнюю границы распределения $p[c \geqslant L]$ числа операций при декодировании c. Эти границы показывают, что рассматриваемое распределение является распределением Парето.

6.4.4. Связь с вероятностью переполнения буфера

Установить связь между распределением $p[c \geqslant L]$ и вероятностью переполнения буфера точно очень трудно. Поэтому здесь сначала обсудим эту связь качественно, а затем покажем, как, следуя подобным качественным рассуждениям, можно получить нижнюю границу для вероятности переполнения буфера.

Операции, которые выполняются декодером в узлах неправильного подмножества кодового дерева, обычно являются опе-

рациями, связанными с обследованием коротких путей этого неправильного подмножества. Действительно, длинные пути имеют цены, значительно отличающиеся от цен правильного пути. В силу этого вероятность того, что выполняемые декодером операции связаны с обследованием длинного пути, является очень малой. Это позволяет предположить, что операции, выполняемые декодером в неправильных подмножествах кодового дерева, являются операциями, которые, как правило, декодер выполняет в небольших локальных «впадинах», образующихся на траектории правильного пути. Поэтому, если некоторая впадина правильного пути содержит $N_{\text{ср}}$ неправильных подмножеств, то суммарное число операций, выполняемых декодером при обследовании этой впадины, является суммой чисел операций, выполняемых декодером в каждом из неправильных подмножеств, а поэтому естественно предположить, что эта сумма имеет то же распределение, что и число операций, выполняемых декодером в каждом из упомянутых выше неправильных подмножеств. Обозначим через c_T суммарное число операций, необходимых для обследования каждой впадины, и пусть c — число операций, выполняемых декодером в одном неправильном подмножестве кодового дерева. Тогда в силу вышеизложенного при больших значениях c и c_T имеем

$$p[c_T \geqslant L] \simeq p[c \geqslant L/N_{\text{ср}}]. \tag{6.62}$$

Предположим, что пики числа операций, обусловленные этими неправильными подмножествами, появляются независимо. Пусть $p_{\text{буф}}(N)$ — вероятность того, что буфер переполнится при декодировании N-го ребра или ранее, а L_0 — число операций, необходимых для заполнения пустого буфера. Тогда, так как всего образуется $N/N_{\text{ср}}$ пиков числа операций, то

$$p_{\text{буф}}(N) \simeq \frac{N}{N_{\text{ср}}} p[c \geqslant L_0/N_{\text{ср}}]. \tag{6.63}$$

Обозначим через σ число операций, выполняемых декодером за время получения одного ребра, и через B — объем регистра в ребрах. Тогда

$$L_0 = \sigma B$$

и, следовательно,

$$p_{\text{буф}}(N) \simeq \frac{N}{N_{\text{ср}}} p[c \geqslant \sigma B/N_{\text{ср}}]. \tag{6.64}$$

Приведенные выше качественные рассуждения подтверждаются результатами моделирования. Кроме того, теоретическим подтверждением правильности этих качественных рассуждений является нижняя граница для $p_{\text{буф}}(N)$, полученная с их помощью Джекобсом и Берлекэмпом [12] на основании оценки

(6.50). В оставшейся части этого раздела остановимся на этой нижней границе.

Предположим, что емкость памяти декодера такова, что в ней могут храниться все данные, необходимые для декодирования θ двоичных информационных символов. Тогда, если за время поступления $u+\theta$ двоичных информационных символов по крайней мере u символов оказались не декодированными, то возникает переполнение буфера. За время поступления $u+\theta$ двоичных информационных символов декодер может выполнить не более чем $\sigma(u+\theta)$ операций. Следовательно, всегда, когда для декодирования u двоичных информационных символов необходимо выполнить более чем $\sigma(u+\theta)$ операций, возникает переполнение буфера. Пусть

$$L=\sigma(u+\theta) \tag{6.65}$$

и u — число двоичных информационных символов, соответствующих N символам канала. Тогда при любой скорости передачи $R \geqslant R_v$ имеет место следующее равенство:

$$u=[NR/\ln 2]^+, \tag{6.66}$$

где u — целое число, удовлетворяющее соотношениям (6.41) и (6.42) (через $[x]^+$ ($[x]^-$) здесь и ниже обозначается минимальное (максимальное) целое число, большее (меньшее) или равное x).

Далее рассмотрим случай декодирования Z двоичных информационных символов. Эти символы можно разбить на $[Z/u]^-$ подблоков по u двоичных символов в каждом (N символов канала). Предположим, что декодер имеет учителя, и всякий раз, когда он начинает декодирование подблока неправильного пути, учитель указывает на это. Декодер возвращается по этому подблоку назад, начинает обследование следующего пути и, таким образом, в конце концов обнаруживает правильный подблок. Другими словами, предположим, что декодер с учителем всегда следует по $[Z/u]^-$ подблокам правильного пути. Таким образом, в случае декодера с учителем вероятность ω возникновения переполнения буфера во время декодирования любого из $[Z/u]^-$ подблоков ограничена снизу вероятностью того, что для обнаружения правильного пути для этого подблока потребуется $\sigma(\theta+u)$ или более операций, т. е.

$$\omega \geqslant Q_N(L). \tag{6.67}$$

Так как в данном случае рассматривается канал без памяти, то шумы, действующие на тот или иной фиксированный подблок, не зависят от шумов, действующих на другие подблоки. Поскольку вероятность ошибки при обработке Z двоичных информационных символов равна нулю для декодера с учителем,

то, используя разложение бинома, для вероятности $p_G(Z)$ переполнения буфера для одного подблока получаем следующую оценку:

$$p_G(Z) = 1 - (1-\omega)^{\lfloor Z/u \rfloor^-} \geqslant [Z/u]^- \omega - \binom{[Z/u]^-}{2} \omega^2 \geqslant$$
$$\geqslant (Z/u - 1)\,\omega\,(1 - Z\omega/2u). \qquad (6.68)$$

Далее рассмотрим случай декодера без учителя. Всякий раз, когда декодер с учителем переполняет буфер, декодер без учителя либо также переполняет буфер, либо возникает ошибка. (При отсутствии ошибки декодер без учителя выполняет все операции, которые выполняет декодер с учителем, и, кроме того, при выходе за пределы начальных подблоков неправильных путей он должен также выполнять операции на последующих подблоках. Эти операции выполняют функции учителя и при отсутствии ошибок указывают декодеру на то, что он вышел из подблока неправильного пути.) Следовательно, вероятность переполнения буфера декодером с учителем является границей сверху для вероятности переполнения буфера декодером без учителя или возникновения ошибки, т. е.

$$p_0(Z) \geqslant p_G(Z). \qquad (6.69)$$

Следующее неравенство следует непосредственно из (6.42):

$$\ln L > (N - X)\,(R_\nu - \hat{E}_0'(\nu)) - O_1\left(\sqrt{N - X}\right). \qquad (6.70)$$

Из формул (6.65) и (6.66) для $\ln L$ получаем следующую оценку:

$$\ln L = \ln\{\sigma\theta + \sigma\,[NR/\ln 2]^+\} \leqslant \ln\sigma\theta + \ln\left\{1 + \frac{1}{\theta}\left(\frac{NR}{\ln 2} + 1\right)\right\}.$$

(Здесь мы воспользовались также тем, что $[NR/\ln 2]^+ \leqslant$ $\leqslant (NR/\ln 2) + 1$.) Разлагая второе слагаемое в правой части последнего неравенства, получаем

$$\ln L \leqslant \ln\sigma\theta + (NR/\theta \ln 2) + 1/\theta. \qquad (6.71)$$

Из формул (6.69) и (6.71) непосредственно следует, что

$$(N - X)\,[R_\nu - \hat{E}_0'(\nu)] - O_1\left(\sqrt{N - X}\right) < \ln L \leqslant$$
$$\leqslant \ln\sigma\theta + (NR/\theta \ln 2) + 1/\theta.$$

Объединяя слагаемые, содержащие N, имеем

$$N\,(R_\nu - \hat{E}_0'(\nu) - R/\theta \ln 2) - O_1\left(\sqrt{N - X}\right) \leqslant$$
$$\leqslant \ln\sigma\theta + X\,\{R_\nu - \hat{E}_0'(\nu)\} + 1/\theta. \qquad (6.72)$$

Отсюда следует, что если

$$\theta > R/\{R_v - \hat{E}_0'(v)\} \ln 2, \tag{6.73}$$

то

$$N = O_5(\ln \sigma\theta). \tag{6.74}$$

Заметим, что условие (6.73) в большинстве практически интересных случаев выполняется.

Согласно формулам (6.66) и (6.73),

$$u = [NR/\ln 2]^+ = O_6(\ln \sigma\theta). \tag{6.75}$$

Отсюда, а также из формул (6.52), (6.65), неравенства $\omega \geqslant Q_N(L)$ и формулы (6.75) получаем

$$\begin{aligned}
\omega &\geqslant L^{-v} \exp\{-O_A(\ln L)^{-1/2}\} = \\
&= (\sigma\theta)^{-v}\left(1 + \frac{u}{\theta}\right)^{-v} \exp\left\{-O_4\sqrt{\ln \sigma\theta + \left[1 + \frac{u}{p}\right]}\right\} = \\
&= (\sigma\theta)^{-v} \exp\left\{-v \ln\left(1 + \frac{O_6(\ln \sigma\theta)}{\theta}\right) - \right. \\
&\quad \left. - O_4\sqrt{\ln \sigma\theta + \ln\left[1 + \frac{O_6(\ln \sigma\theta)}{\theta}\right]}\right\} \geqslant \\
&\geqslant (\sigma\theta)^{-v} \exp\{-O_7\sqrt{\ln \sigma\theta}\}.
\end{aligned} \tag{6.76}$$

Предположим, что ω и Z таковы, что $Z > 2u = 2 \times O_6(\ln \sigma\theta)$ и $(\omega Z/2u) \leqslant 1/2$. При этих предположениях из формул (6.68) и (6.69) получаем

$$\begin{aligned}
p_0(Z) \geqslant p_G(Z) &\geqslant \left(\frac{Z}{u} - 1\right) u\left(1 - \frac{Z\omega}{2u}\right) \geqslant \frac{\omega Z}{2u}\left(1 - \frac{Z\omega}{2u}\right) \geqslant \frac{Z\omega}{4u} \geqslant \\
&\geqslant Z(\sigma\theta)^{-v} \exp\{-O_7\sqrt{\ln \sigma\theta} - \ln[4O_6(\ln \sigma\theta)]\} \simeq \qquad (6.77) \\
&\simeq Z(\sigma\theta)^{-v} \exp\{-O_8\sqrt{\ln \sigma\theta}\}. \qquad (6.78)
\end{aligned}$$

Таким образом, сравнительно просто была получена нижняя граница для вероятности ошибки или переполнения буфера. Этот результат не является столь же сильным, как верхняя граница. Нижняя граница и верхняя граница, полученная с помощью качественных рассуждений, похожи друг на друга. Из формул (6.64) и (6.57) следует, что

$$\begin{aligned}
P_{\text{буф}}(N) &\simeq \frac{N}{N_{\text{ср}}}\, p[c \geqslant \sigma B/N_{\text{ср}}] \leqslant \frac{N}{N_{\text{ср}}}\, G(\sigma B/N_{\text{ср}})^{-R_{\text{выч}}/R} = \\
&= \frac{N}{N_{\text{ср}}^{1-(R_{\text{выч}}/R)}}\, G(\sigma B)^{-R_{\text{выч}}/R}.
\end{aligned} \tag{6.79}$$

Полагая в нижней границе $Z = N$, $\theta = B$, $\nu = R_{\text{выч}}/R$, получаем

$$p_0(N) \geqslant \frac{N}{u}(\sigma B)^{-R_{\text{выч}}/R} G \exp\{-O_7(\sqrt{\ln \sigma B})\} \simeq$$

$$\simeq \frac{N}{u} G (\sigma B)^{R_{\text{выч}}/R}. \quad (6.80)$$

Таким образом, вероятность переполнения буфера линейно возрастает с N и убывает с ростом σB.

6.5. Вероятность необнаружения ошибки

При исследовании среднего числа операций для алгоритма Фано использовалось предположение о бесконечности кодового ограничения сверточного кода, но полученная в результате верхняя граница для среднего числа операций оказалась независящей от длины кодового ограничения, и это позволило сделать вывод о том, что предположение о бесконечности кодового ограничения допустимо.

Однако кодовое ограничение сверточных кодов, порождаемых реальными кодерами, конечно и поэтому могут возникать необнаруживаемые ошибки. Например, вернемся к показанному на фиг. 6.1 сверточному кодеру кода со скоростью $1/2$. Этот кодер содержит 5-разрядный регистр сдвига, и информационный символ, поступающий в момент времени i, еще влияет на проверочный символ, формируемый в момент времени $i + 5$, но уже не оказывает влияния на последующие проверочные символы. В силу этого две выходные последовательности кодера, соответствующие двум двоичным входным последовательностям, отличающимся в первых s символах и совпадающим в остальных символах, могут отличаться лишь в первых $2(s + 5)$ символах, но обязательно совпадают в последующих символах. Предположим, что при передаче одной из этих выходных последовательностей по каналу шумы исказили первые $2(s + 5)$ символов таким образом, что эта выходная последовательность перешла в другую указанную выше последовательность. В этом случае возникшую ошибку декодер не может ни обнаружить, ни исправить. Такие ошибки называют необнаруживаемыми ошибками.

Известно несколько границ вероятности ошибки для алгоритма Фано. В частности, для всех скоростей, меньших пропускной способности канала, верхняя граница для вероятности ошибки была получена Юдкиным [13]. Однако здесь мы рассмотрим верхнюю границу для вероятности ошибки, полученную Сэваджем [9] из соотношений, приведенных в разд. 6.3.

Вообще говоря, двоичная последовательность длины $k_0 s$, поступающая на вход сверточного кодера, содержащего k_0 m-раз-

рядных регистров сдвига, порождает s ребер и оказывает влияние на формирование $t = s + m + 1$ ребер. Предположим, что задан некоторый фиксированный правильный путь, и попробуем найти число $\tau(s)$ путей в первом неправильном подмножестве, отличающихся от заданного правильного пути в первых ребрах и совпадающих с ним в последующих ребрах. Сначала заметим, что из каждого узла кодового дерева выходит $b = 2^{k_0}$ ребер. Чтобы путь принадлежал первому неправильному подмножеству, его первое ребро должно отличаться от первого ребра правильного пути. При этом для того, чтобы путь отличался от правильного пути самое большее в s ребрах, его s-е ребро должно отличаться от s-го ребра правильного пути. Отсюда следует, что

$$\tau(s) = (2^{k_0} - 1)(2^{k_0})^{s-2} \leqslant 2^{sk_0}. \tag{6.81}$$

В случае если путь U_t длины $t = s + m + 1$ ребер, лежащих в неправильном подмножестве, начинающемся в начальном узле кодового дерева, имеет цену, превышающую максимальное значение порога, которое лежит ниже минимальной цены правильных путей длины tn_0 или менее символов, декодер Фано может по ошибке идентифицировать путь U_t как правильный и породить таким образом s или менее ошибок. Другими словами, если путь U_t такой, что

$$\gamma(U_t, V_t) - \Big(\min_{0 \leqslant w \leqslant t} \gamma(U_w^*, V_w) - D_0\Big) \geqslant 0, \tag{6.82}$$

то декодер Фано может декодировать этот путь ошибочно. Следовательно, вероятность $p_e(s \mid U^*)$ возникновения s или менее необнаруживаемых ошибок при передаче правильного пути можно найти, используя почти те же рассуждения, что и в разд. 6.3. Пусть $W(x)$ — ступенчатая функция, определенная выше [см. формулу (6.12)]. Тогда

$$p_e(s \mid U_t^*) \leqslant \sum_{V_t} p(V_t^* \mid U_t^*) \sum_{m=1}^{\tau(s)} \sum_{w=0}^{t} W[\gamma(U_t, V_t) - \gamma(U_w^*, V_w^*) + D_0]. \tag{6.83}$$

Далее, точно так же, как и в разд. 6.3, воспользуемся границей Чернова

$$W(x) \leqslant e^{\lambda x}, \quad \lambda > 0, \tag{6.84}$$

подставим

$$\gamma(U_t, V_t) = I(U_t, V_t) - tn_0 U, \tag{6.85}$$

$$\gamma(U_w^*, V_w) = I(U_w^*, V_w) - wn_0 U \tag{6.86}$$

в правую часть (6.83) и проведем усреднение по ансамблю кодов. Здесь следует заметить, что в разд. 6.3 допускались бесконечные кодовые ограничения, а среднее значение по ансамблю

вычислялось для ансамбля кодов бесконечно большой длины. В данном же случае кодовое ограничение кода n_A равно $(m+1)n_0$, а длина кодов в ансамбле, по которому производится усреднение, составляет $(t+m+1)n_0 = tn_0 + n_A$ символов. Если на длине $tn_0 + n_A$ символов кодовые последовательности из этого ансамбля статистически независимы, то в других местах эти последовательности оказываются, вообще говоря, уже статистически зависимыми. Как показал Галлагер [11], ансамбль кодов, обладающих этими свойствами, можно построить, выбирая всякий раз, когда в регистры сдвига кодера поступают новые информационные символы, случайным образом связи и осуществляя сложение выходной последовательности кодера с некоторой случайной последовательностью.

Теорема 6.3. *Для любого $R < R_{\text{выч}}$ существует сверточный код со скоростью передачи R и вероятностью необнаружения ошибки, экспоненциально убывающей с ростом длины кодового ограничения n_A. Показатель экспоненты этой вероятности меньше чем $R_{\text{выч}}$.*

Доказательство. Подставляя (6.84)—(6.86) в (6.83), получаем

$$p_e(s \mid U^*) \leqslant \sum_{m=1}^{\tau(s)} \sum_{w=0}^{t} \sum_{V_t} p(V_t \mid U_t^*) \exp\{I(U_t, V_t) -$$

$$- I(U_w^*, V_w) - tn_0U + wn_0U + D_0\} =$$

$$= \sum_{m=1}^{\tau(s)} \sum_{w=0}^{t} \sum_{V_t} p(V_t \mid U_t^*)\, e^{\lambda D_0} e^{\lambda (w-t) n_0 U} \left[\frac{p(V_t \mid U_t)}{Q(V_t)}\right]^{\lambda} \left[\frac{Q(V_w)}{p(V_w \mid U_w)}\right]^{\lambda} =$$

$$= \sum_{r=0}^{t} \sum_{m=1}^{\tau(s)} \sum_{V_t} e^{\lambda D_0} e^{\lambda (w-t) n_0 U} \left[\frac{[p(V_t \mid V_t)]^{\lambda}}{[Q(V_{t-w})]^{\lambda}}\right] \times$$

$$\times [p(V_w \mid U_w^*)]^{1-\lambda} [p(V_w \mid U_w^*)^{1-\lambda}] [p(V_{t-w} \mid U_{t-w}^*)]. \quad (6.87)$$

Проводя усреднение по ансамблю кодов, полагая

$$[E_{ns}]_{\text{ср}} =$$

$$= \sum_{V_t} \overline{p(V_{t-w} \mid U_{t-w}^*)} \overline{[p(V_w \mid U_w^*]^{1-\lambda}}\, \overline{[p(V_t \mid U_t)]^{\lambda}}\, \overline{[Q(V_{t-w})]^{\lambda}} \quad (6.88)$$

и используя соотношения $\overline{p(V_{t-w} \mid U_{t-w}^*)} = Q(V_{t-w}) \leqslant 1$, получаем

$$[E_{ns}]_{\text{ср}} \leqslant \sum_{V_t} \overline{[p(V_w \mid U_w^*)]^{-\lambda}}\, \overline{[p(V_t \mid U_t)]^{\lambda}}. \quad (6.89)$$

Согласно неравенству Шварца,

$$[E_{ns}]_{\text{ср}} \leqslant \left[\sum_{V_t}\left\{[p(V_w \mid U_w^*)]^{1-\lambda}\right\}^2\right]^{1/2}\left[\sum_{V_t}\left\{[p(V_t \mid U_t)]^{\lambda}\right\}^2\right]^{1/2}. \quad (6.90)$$

Далее, вводя $R_{\text{выч}}$ с помощью равенств (6.30) и (6.31) и проводя суммирование по m, находим

$$p_e(s \mid U^*) \leqslant \left(\sum_{w=0}^{t} e^{\lambda w n_0 U} e^{-\lambda w n_0 R_{\text{выч}}}\right) e^{sk_0 \ln 2} e^{\lambda D_0} e^{-\lambda_t (n_0 + R_{\text{выч}})}. \quad (6.91)$$

Если это неравенство усреднить по U^*, то получим границу для средней вероятности ошибки $p_e(s)$. Однако правая часть (6.91) не зависит от U^* и, следовательно, (6.91) дает верхнюю границу также для $p_e(s)$. Вводя величины $t = s + m + 1$, $R = k_0/n_0$, $n_A = n_0(m+1)$, получаем

$$p_e(s) \leqslant \left(\sum_{w=0}^{s+m+1} e^{\lambda w n_0 (U - R_{\text{выч}})}\right) e^{sn_0 (R \ln 2 - \lambda U - \lambda R_{\text{выч}})} e^{-(n_A - n_0)\lambda (U + R_{\text{выч}})} \quad (6.92)$$

Вероятность p_e возникновения по крайней мере одной необнаруживаемой ошибки в неправильном подмножестве, начинающемся в начальном узле кодового дерева, ограничена сверху суммой по $\bar{s}$ вероятностей $p_s(s)$:

$$p_e \leqslant \sum_{s=1}^{\infty} p_e(s) =$$

$$= \sum_{s=1}^{\infty}\left(\sum_{w=0}^{s+m+1} e^{\lambda n_0 w (U - R_{\text{выч}})}\right) e^{sn_0 (R \ln 2 - \lambda U - \lambda U_{\text{выч}})} e^{-(n_A - n_0)\lambda (U + R_{\text{выч}})} e^{\lambda D_0}. \quad (6.93)$$

Правая часть этого неравенства сходится, если $U < R_{\text{выч}}$ и $R < (\lambda(U + R_{\text{выч}})/\ln 2)$. В случае, если $R > R_{\text{выч}}$, то, как видно из разд. 6.3, среднее число операций расходится. Если положить $\lambda = (\ln 2)2$, то при $R < R_{\text{выч}}$ правая часть неравенства (6.93) будет сходиться. Если в этой области значений R суммирование по w и s в правой части (6.93) провести независимо, положив верхние пределы равными бесконечности, то получим следующее неравенство:

$$p_e \leqslant \left\{1 - e^{n_0 (\ln 2/2)(U - R_{\text{выч}})}\right\}^{-1}\left\{1 - e^{n_0 \ln 2 (R - U/2 - R_{\text{выч}}/2)}\right\}^{-1} \times$$
$$\times e^{(\ln 2/2)(D_0 + 2R_{\text{выч}})} e^{-n_A \ln 2/2)(U + R_{\text{выч}})}. \quad (6.94)$$

Обозначив через A произведение первых трех сомножителей в правой части (6.94), последнее неравенство перепишем в виде

$$p_e \leqslant A e^{-n_A \ln 2/2 (U + R_{\text{выч}})} \leqslant A e^{-n_A \ln 2 R_{\text{выч}}}. \quad (6.95)$$

Отсюда следует справедливость доказываемой теоремы.

Таким образом, для алгоритма Фано в случае $R < R_{\text{выч}}$ были найдены границы для вероятности ошибки, среднего числа операций и вероятности переполнения буфера. Чтобы получить представление о порядке этих параметров, рассмотрим следующий пример.

Пример 6.1. Рассмотрим двоичный симметричный канал с вероятностью ошибки $p = 0{,}01$ и сверточный код со скоростью передачи $R = 1/2 = 0{,}974R_{\text{выч}}$ и кодовым ограничением $n_A = 120$. Пусть $U = 0{,}974R_{\text{выч}}$ и $T_0 = 2R_{\text{выч}}$. При $p_1 = p_2 = 1/2$ и $\nu = 1$ получаем

$$R_{\text{выч}} = \nu \ln 2 - (1+\nu)\ln\left[0{,}99^{1/(1+\nu)} + 0{,}01^{1/(1+\nu)}\right] = 0{,}512.$$

Верхняя граница для вероятности ошибки имеет следующий вид:

$$p_e \leqslant \left\{1 - e^{-0.026 \ln 2 R_{\text{выч}}}\right\}^{-1} \left\{1 - e^{\ln 2\,(1-1{,}974R_{\text{выч}})}\right\}^{-1} \times$$
$$\times\, e^{R_{\text{выч}} \ln 2} e^{-60 \cdot \ln 2 \cdot 1{,}974 R_{\text{выч}}} \simeq 8 \cdot 10^{-17}.$$

Пусть $\sigma = 20$ и $B = 1000$. Полагая $\exp\left\{-O_8\sqrt{\ln \sigma B}\right)\} \simeq$ $\simeq \exp\left\{-\sqrt{\ln \sigma B}\right\}$, $\nu \simeq R_{\text{выч}}/R$, из формулы (6.78) находим вероятность переполнения буфера

$$p_0(N) \simeq \frac{N}{460\,000}.$$

При передаче со скоростью 2400 бит/с переполнение буфера возникает в среднем один раз за $460\,000/2400 \simeq 200$ с. В то же время ошибка появляется в среднем один раз за 10^{14} с.

Как показывают приведенные выше численные результаты, поведение алгоритмов последовательного декодирования определяется главным образом вероятностью переполнения буфера.

6.6. Границы Витерби и декодирование по максимуму правдоподобия

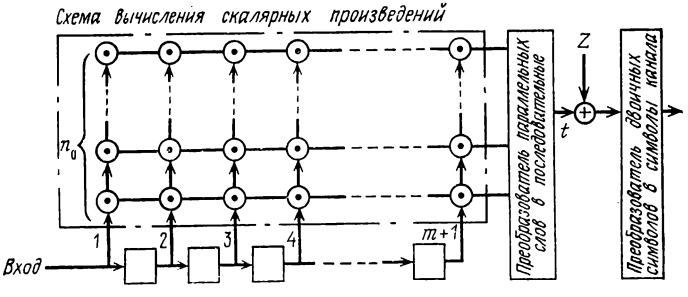
Начало этой главы, за исключением вывода нижней границы для вероятности переполнения буфера, посвящено исследованию вероятности ошибки, среднего числа операций и распределения числа операций для алгоритма последовательного декодирования, предложенного Фано. В этом разделе находятся верхняя и нижняя границы для вероятности ошибки при декодировании методом Витерби, который не связан с алгоритмом Фано. Верхняя граница будет получена с помощью вероятностного (но не последовательного) алгоритма декодирования, предложенного Витерби, а вывод нижней границы вообще не связан с каким-либо конкретным алгоритмом декодирования, и нижняя

граница справедлива для произвольного сверточного кода. Полученные ниже границы Витерби показывают, что характеристики сверточных кодов лучше, чем у блоковых кодов той же длины. Это утверждение неоднократно высказывалось после введения Элайсом сверточных кодов, но строгое доказательство оно получило лишь с появлением границ Витерби.

Форни [14] показал, что использованный Витерби для получения верхней границы алгоритм декодирования является алгоритмом декодирования по максимуму правдоподобия, и таким образом еще раз отметил его важность.

6.6.1. Верхняя граница для вероятности ошибки

Верхняя граница Витерби так же, как и полученная выше верхняя граница вероятности ошибки для алгоритма Фано, выводится методом случайного кодирования. Использованный в



Фиг. 6.13. Кодер.

разд. 6.6 ансамбль сверточных кодов можно построить также следующим образом.

Входная последовательность, символами которой являются элементы поля $GF(2)$, как показано на фиг. 6.13, подается на вход m-разрядного регистра сдвига, входящего в состав кодера. С каждым поступлением в регистр сдвига очередного входного символа с помощью схемы вычисления скалярных произведений формируется $n_0 t$ выходных символов. Схема вычисления скалярных произведений выбирается случайным образом с помощью ансамбля $2^{(m+1)\,n_0}$ матриц с равномерным распределением [1]). Выходная последовательность t складывается со слу-

[1]) На фиг. 6.13 каждый элемент, обозначенный кружком с точкой внутри, выполняет умножение символа из соответствующего разряда регистра на двоичный символ, который для этого элемента выбирается случайно, независимо и равновероятно среди {0, 1}. Все произведения вдоль каждой горизонтальной строки складываются и подаются на соответствующий вход преобразователя параллельных слоев в последовательные. — *Прим. ред.*

чайной последовательностью Z независимых равновероятных символов. В результате все символы на выходе устройства формирования символов канала оказываются равновероятными. После того как на вход кодера будет подано B информационных символов, сюда подается последовательность из $m+1$ нулей. Порождаемый таким кодером код обладает следующими свойствами:

1. Если $m+1$ информационных символов двух различных путей кодового дерева совпадают, эти информационные символы порождают одну и ту же последовательность из n_0 символов.

2. Все символы канала вдоль любого пути кодового дерева являются равновероятными.

3. Символы канала вдоль любого пути кодового дерева статистически независимы.

4. Если с вводом в регистр сдвига каждого информационного символа матрица из $(m+1)n_0$ элементов, используемых для вычисления скалярных произведений, выбирается случайно и независимо, то символы канала различных двух произвольных путей кодового дерева оказываются независимыми.

Поскольку каждому информационному символу описанный выше кодер сопоставляет n_0 символов канала, то порождаемые этим кодером коды имеют скорость передачи

$$R = \ln 2/n_0 \tag{6.96}$$

и кодовое ограничение $n_A = (m+1)n_0$.

Далее, как было сказано, будем предполагать, что после введения в регистр сдвига B информационных символов в него вводятся $m+1$ нулей. При выводе верхней границы для вероятности ошибки будет использоваться следующий алгоритм декодирования.

Каждое ребро заданного пути кодового дерева будет обозначаться посредством соответствующего информационного символа u_i, $u_i \in GF(2)$, а соответствующие этому ребру символы принятой последовательности — через v_i.

1) Для каждого из 2^{m+1} путей кодового дерева, образованных первыми $m+1$ ребрами, вычисляются значения 2^{m+1} функций правдоподобия $\prod_{i=1}^{m+1} p(v_i \mid u_i)$. Далее проводится сравнение значений функций правдоподобия путей

$$\begin{aligned} &(0, u_2, \ldots, u_{m+1}), \\ &(1, u_2, \ldots, u_{m+1}) \end{aligned} \tag{6.97}$$

для всех 2^m возможных векторов $(u_2, \ldots, u_{m+1})$. При проведении каждого из этих 2^m сравнений путь, имеющий наиболь-

шее значение функции правдоподобия, оставляется, а другой путь отбрасывается. После выполнения 2^m сравнений остается 2^m векторов, и для каждого из них вектор $(u_2, u_3, \ldots, u_{m+1})$ определяется однозначно.

2) Для каждого из оставленных на шаге 1 путей вычисляются значения функций правдоподобия двух ребер, выходящих из $(m+2)$-го узла кодового дерева; полученные значения умножаются на значение функции правдоподобия исходного пути, и таким образом находятся 2^m значений функций правдоподобия путей, состоящих из $m+2$ ребер. Эти последовательности $(u_2, u_3, \ldots, u_{m+2})$ определяются однозначно. Пусть $\mu_{w1}^{(1)}$ $(w = 1, 2)$ — символ u_1 последовательности, оставленной на 1-м шаге. Для различных пар путей

$$\begin{aligned}&(\mu_{11}^{(1)}, 0, u_3, u_4, \ldots, u_{m+2}),\\ &(\mu_{21}^{(1)}, 1, u_3, u_4, \ldots, u_{m+2})\end{aligned} \tag{6.98}$$

находятся 2^{m+1} значений функций правдоподобия. Далее, как и на 1-м шаге, 2^m из этих путей оставляются, а остальные отбрасываются.

Этот шаг процедуры декодирования повторяется $B - m$ раз. В частности, на $(w+1)$-м шаге декодер выполняет 2^m сравнений функций правдоподобия для следующих пар путей:

$$\begin{aligned}&(\mu_{11}^{(w)}, \mu_{12}^{(w)}, \ldots, \mu_{1w}^{(w)}, 0, u_{w+2}, \ldots, u_{w+m+1}),\\ &(\mu_{21}^{(w)}, \mu_{22}^{(w)}, \ldots, \mu_{2w}^{(w)}, 1, u_{w+2}, \ldots, u_{w+m+1}),\end{aligned} \tag{6.99}$$

где $\mu_{k1}^{(w)}, \ldots, \mu_{kw}^{(w)}$ $(k = 1, 2)$ — символы, полученные в результате сравнений на предыдущих шагах декодирования.

3) На $(B - m + 1)$-м шаге на вход декодера подается символ 0, проводится сравнение значений функций правдоподобия следующих 2^{m-1} пар путей:

$$\begin{aligned}&(\mu_{11}^{(B-m)}, \ldots, \mu_{1,B-m}^{(B-m)}, 0, u_{B-m+2}, \ldots, u_B, 0),\\ &(\mu_{21}^{(B-m)}, \ldots, \mu_{2,B-m}^{(B-m)}, 1, u_{B-m+2}, \ldots, u_B, 0)\end{aligned} \tag{6.100}$$

при всевозможных векторах $u_{B-m+2}, \ldots, u_B$ и по результатам этих сравнений оставляются 2^{m-1} путей. Так как на вход регистра сдвига с момента $B+1$ подаются нулевые символы, то число выполняемых сравнений с каждым шагом уменьшается вдвое. Перед B-м шагом остаются два пути:

$$\begin{aligned}&(\mu_{11}^{(B-1)}, \ldots, \mu_{1,B-1}^{(B-1)}, 0, 0, \ldots, 0),\\ &(\mu_{21}^{(B-1)}, \ldots, \mu_{2,B-1}^{(B-1)}, 1, 0, \ldots, 0).\end{aligned} \tag{6.101}$$

На B-м шаге из этих двух путей выбирается тот, который имеет большее значение функции правдоподобия, и этот путь оставляется как правильный.

При использовании этого алгоритма декодирования ошибка при декодировании возникает в том случае, если при проведении B сравнений для пар путей, содержащих правильный путь, оказывается, что значение функции правдоподобия правильного пути меньше, чем значение функции правдоподобия некоторого другого сравниваемого пути. В целях упрощения предположим, что правильный путь является последовательностью, состоящей из одних нулей. Рассмотрим случай, когда ошибка возникает на $(w+1)$-м шаге декодирования. На $(w+1)$-м шаге декодирования первые $w+1$ ребер правильного пути обозначим через $(\mathbf{0}) = (0, 0, \ldots, 0)$. Векторы, сравнение которых проводится на этом шаге декодирования, задаются формулами (6.99), но правильный путь входит лишь в следующие пары:

$$\begin{aligned}&(0, 0, \ldots, 0, 0, 0, \ldots, 0),\\ &(\mu_{21}^{(w)}, \mu_{22}^{(w)}, \ldots, \mu_{2w}^{(w)}, 1, 0, \ldots, 0),\end{aligned} \tag{6.102}$$

где $\mu_{2u}^{(w)}$, $u = 1, \ldots, w$, — символы, найденные на предыдущих шагах и являющиеся элементами поля $GF(2)$. Следовательно, число путей, которые сравниваются на $(w+1)$-м шаге с правильным путем, равно 2^w. Эти 2^w путей можно разбить на группы следующим образом.

Пусть $(\underbrace{0, \ldots, 0}_{w}, 1, \underbrace{0, \ldots, 0}_{m})$ — вектор, обозначающий путь, ответвляющийся от правильного пути в $(w+1)$-м узле, а $(\underbrace{0, \ldots, 0}_{w-1}, 1, 1, \underbrace{0, \ldots, 0}_{m})$ — вектор, обозначающий путь, ответвляющийся от правильного пути в w-м узле. По аналогии пусть $(\underbrace{0, \ldots, 0}_{w-2}, 1, \delta, 1, \underbrace{0, \ldots, 0}_{m})$ —вектор, обозначающий путь, ответвляющийся от правильного пути в $(w-1)$-м узле $(\delta = 0, 1)$; всего имеются два таких пути. В общем случае число путей, ответвляющихся от правильного пути в $(w+1-k)$-м узле, равно 2^{k-1} $(k = 1, \ldots, w)$, а число путей, ответвляющихся в $(w+1)$-м узле, равно 1. Общее число указанных выше путей равно 2^w, и оно является общим числом векторов, которые могут составлять вторую строку (6.102).

Однако все эти векторы не имеют прямого отношения к выбору правильного пути. В силу свойства 1 кодера, если два пути, отличающиеся в некотором заданном ребре, совпадают в последующих $m+1$ информационных символах, то они также совпадают в следующем за ними ребре. Если два пути с про-

извольными фиксированными информационными символами не имеют никаких $m+1$ совпадающих подряд идущих информационных символов, то они не имеют совпадающих ребер. Такие пути будем называть полностью несовпадающими. В силу этого свойства кодера при рассматриваемом алгоритме декодирования векторами, которые сравниваются с правильным путем на $(w+1)$-м шаге, являются не все векторы вида (6.99), а только те из них, которые полностью не совпадают с правильным путем. Следовательно, математическое ожидание вероятности ошибки при проведении сравнений на $(w+1)$-м шаге можно оценить сверху суммой математических ожиданий вероятностей возникновения ошибок при проведении сравнений с векторами, отличающимися от правильного пути соответственно в $(w+1)$-м, ..., 1-м ребрах:

$$\overline{p(w+1)} < \sum_{k=0}^{w} \Pr\left\{\begin{array}{l}\text{ошибка происходит из-за пути,}\\ \text{ответвляющегося от правильного}\\ \text{на } (w+1-k)\text{-м ребре}\end{array}\right\}. \quad (6.103)$$

Слагаемое с $k=0$ в правой части этого неравенства можно оценить сверху вероятностью ошибки для блокового кода длиной $n_A=(m+1)n_0$, состоящего из одного кодового слова; k-е слагаемое в правой части (6.103) можно оценить сверху вероятностью ошибки для блокового кода длиной $n_A+(\ln 2/R)k$ символов канала, состоящего из 2^{k-1} кодовых слов. Используя свойства 2—4 кодера и границу случайного кодирования, получаем

$$\overline{p(w+1)} < \exp[-n_A E_0(\nu)] + \sum_{k=1}^{w} \{2^{k-1}\}^{\nu} \exp\left[-\left(n_A + \frac{\ln 2}{R}k\right)E_0(\nu)\right] <$$

$$< \exp[-n_A E_0(\nu)] \sum_{k=0}^{\infty} 2^{k[\nu - E_0(\nu)/R]} = \frac{1}{1-2^{-\varphi/R}} \exp[-NE_0(\nu)], \quad (6.104)$$

где

$$\varphi = E_0(\nu) - \nu R > 0. \quad (6.105)$$

Верхняя граница для вероятности ошибки при декодировании древовидного кода $\bar{p}_E$ длиной B получается суммированием (6.104) по w. Так как правая часть неравенства (6.104) не зависит от w, то имеем

$$\overline{p}_E < \sum_{w=0}^{B-1} \overline{p(w+1)} < \frac{B}{1-2^{-\varphi/R}} \exp[-n_A E_0(\nu)], \quad 0 < \nu \leqslant 1. \quad (6.106)$$

В силу того что в ансамбле кодов, для которого справедливы неравенства (6.106), существует по крайней мере один код,

для которого $p_E < \bar{p}_E$, а $E_0(\nu)$ является монотонно возрастающей функцией ν, получаем следующую теорему.

Теорема 6.4. (Витерби) *В дискретном канале без памяти вероятность ошибки при декодировании B ребер двоичного древовидного кода удовлетворяет следующему неравенству*:

$$p_E < \frac{B}{1-2^{-\varphi/R}} \exp[-n_A E(R)], \tag{6.107}$$

где

$$E(R) = \begin{cases} R_0, & 0 \leqslant R = R_0 - \varphi < R_0, \\ E_0(\nu), & R_0 - \varphi \leqslant R \leqslant \dfrac{E_0(\nu) - C}{\nu} < c, \end{cases} \tag{6.108}$$

$$R_0 = E_0(1) = \max_{Q(u)} \left\{ -\ln \sum_v \left[\sum_u Q(u) \sqrt{p(v|u)} \right]^2 \right\}.$$

6.6.2. Нижняя граница для вероятности ошибки

В этом разделе получим нижнюю границу для вероятности ошибки. Для этого воспользуемся декодером с учителем, для исследования которого применим границу сферической упаковки и прямолинейную границу.

В данном случае будем предполагать, что учитель указывает декодеру все ребра правильного пути кодового дерева, за исключением l следующих друг за другом ребер $w+1$, $w+2$, ..., $w+l$ $(0 \leqslant w \leqslant B-l)$. Другими словами, учитель указывает все передаваемые информационные символы v_i, за исключением l символов, соответствующих указанным выше ребрам. Таким образом, задача декодера сводится к тому, чтобы по принятой последовательности $v = (v_1, v_2, \ldots, v_{B+m})$, состоящей из $(B+m)n_0$ символов, определить, какая из 2^l двоичных последовательностей передавалась. (Здесь v_i — совокупность принятых символов, соответствующих i-му ребру.) Итак, поскольку в данном случае символы, передаваемые в моменты времени $i < w$, даются учителем, то декодеру необходимо исследовать лишь символы v_i, $i \geqslant w+1$. Исследуемые 2^l путей могут различаться лишь в $(w+1)$-м и последующих ребрах, но если рассматривать только информационные последовательности, имеющие одни и те же символы v_i при $i \geqslant w+l+1$, то состояния регистра сдвига при порождении $(w+l+1)$-го и последующих ребер будут совпадать и, следовательно, при $i \geqslant w+l+1$ кодер будет порождать всегда одну и ту же последовательность. Таким образом, исследуемые 2^l путей могут отличаться друг от друга самое большее в $l+m$ ребрах.

Будем считать, что приемник для каждого из этих 2^l путей, имеющих равные априорные вероятности, вычисляет значение

функции правдоподобия $p(v|u)$, $v=(v_{w+1}, \ldots, v_{w+l+m})$, $u=(u_{w+1}, \ldots, u_{w+l})$, и выбирает из них тот путь, для которого значение функции правдоподобия максимально. Здесь u — вектор, определяющий путь. Полагая

$$\eta = l/(m+1), \tag{6.109}$$

из равенства

$$n_A = (m+1)n_0 \tag{6.110}$$

получаем

$$l \ln 2 = \eta n_A R. \tag{6.111}$$

Длина вектора v равна $(\eta+1)n_A + n_0$. Следовательно, вероятность ошибки для указанного выше декодера можно оценить снизу любой нижней границей для вероятности ошибки блокового кода длиной $(\eta+1)n_A - n_0$, содержащего $2^l = e^{\eta n_A R}$ кодовых слов.

Как следует из соотношений (1.59) и (1.63) первой главы, граница сферической упаковки для блокового кода длиной n со скоростью R' имеет следующий вид:

$$p_e \geqslant \exp[-n\{E_{sp}(R' - O_1(n)) + O_2(n)\}], \tag{6.112}$$

$$E_{sp}(R') = \max_{0<\nu<\infty} [\hat{E}_0(\nu) - \nu R'], \tag{6.113}$$

где $\hat{E}_0(\nu)$ — верхняя огибающая $E_0(\nu, Q)$. Так как

$$E_{sp}(R' - O_1(\sqrt{n})) = \max_{0<\nu<\infty} \{\hat{E}_0(\nu) - \nu(R' - O_1(\sqrt{n}))\}$$

и параметр ν здесь является некоторой фиксированной постоянной, то $\nu O_1(\sqrt{n})$ стремится к нулю с ростом n и, следовательно,

$$p_e \geqslant \exp[-n\{E_{sp}(R') + O_3(\sqrt{n})\}]. \tag{6.114}$$

Подставляя $n = (\eta+1)n_A - n_0$, получаем

$$p_{E\eta} \geqslant \exp[-\{(\eta+1)n_A - n_0\}\{E_{sp}(R') + O_3(\sqrt{n_A})\}] \geqslant$$
$$\geqslant \exp[-\{(\eta+1)n_A\}\{E_{sp}(R') + O_3(\sqrt{n_A})\}]. \tag{6.115}$$

Так как аргумент R' функции $E_{sp}(R')$ удовлетворяет неравенству

$$R' \frac{\ln M}{n} = \frac{\eta n_A R}{(\eta+1)n_A - n_0} > \frac{\eta}{\eta+1} R,$$

то

$$E_{sp}(R) < E_{spL}(R, n) = \max_{0 \leqslant \nu < \infty} \left[\hat{E}_0(\nu) - \nu \frac{\eta}{\eta+1} R\right]. \tag{6.116}$$

Как следует из результатов Шеннона, Галлагера и Берлекэмпа [5], функция $E_{spL}(R, \eta)$ может быть задана параметрически:

$$E_{spL}(R, \eta) = \hat{E}_0(\nu) - \nu \hat{E}_0'(\nu), \tag{6.117}$$

$$R = \frac{\eta + 1}{\eta} \hat{E}_0'(\nu), \tag{6.118}$$

где $\eta = l/(m+1)$, а l — любое из чисел $1, \ldots, B$. Проводя максимизацию $p_{E\eta}$ по ν, получаем следующую нижнюю границу для вероятности ошибки p_E:

$$p_E(n_A, R) \geqslant \max_{1/(m+1) \leqslant \eta \leqslant B/(m+1)} p_{E_\eta}(n_A, R) >$$

$$> \exp\{-n_A \min_{1/(m+1) \leqslant \eta \leqslant B/(m+1)} (\eta + 1)[E_{spL}(R, \eta) + O(\eta, \eta_A)]\}. \tag{6.119}$$

Здесь необходимо найти функцию

$$E_{spL}(R) = \min_{1/(m+1) \leqslant \eta \leqslant B/(m+1)} (\eta + 1) E_{spL}(R, \eta). \tag{6.120}$$

Для этого, выбрав $B/(m+1)$ достаточно большим, сначала проведем минимизацию по всем действительным положительным числам η, а после этого ограничим область минимизации для чисел η, кратных $1/(m+1)$.

Дифференцируя $(\eta + 1) E_{spL}(R, \eta)$ по η, получаем

$$\frac{\partial [(\eta+1) E_{spL}(R, \eta)]}{\partial \eta} = \hat{E}_0(\nu) - \nu \hat{E}_0'(\nu) + (\eta + 1)[\nu \hat{E}_0''(\nu)] \frac{\partial \nu}{\partial \eta}. \tag{6.121}$$

Из равенства (6.118) имеем

$$\hat{E}_0''(\nu) \frac{\partial \nu}{\partial \eta} = \frac{\partial}{\partial \eta} \frac{\eta}{\eta + 1} R = \frac{R}{(\eta + 1)^2}. \tag{6.122}$$

Подставляя последнее равенство в формулу (6.121) и приравнивая далее (6.121) к нулю, находим $\eta + 1$:

$$\eta + 1 = \frac{\nu R}{\hat{E}_0(\nu) - \nu \hat{E}_0'(\nu)}. \tag{6.123}$$

Повторно дифференцируя равенство (6.121), используя (6.118) и то, что вторая производная $\hat{E}_0''(\nu)$ выпуклой вверх функции $E_0(\nu)$ отрицательна, получаем

$$\frac{\partial^2 [(\eta+1) E_{spL}(R, \eta)]}{\partial \eta^2} = \frac{-R^2}{(\eta+1)^2 \hat{E}_0''(\nu)} \geqslant 0. \tag{6.124}$$

Следовательно, в точке, определяемой соотношением (6.123), достигается минимум исследуемой функции. Подставляя ν из

формулы (6.123) в (6.118), имеем

$$R(\nu R - \hat{E}_0(\nu)) = 0. \tag{6.125}$$

Отсюда находим R:

$$R = \frac{\hat{E}_0(\nu)}{\nu}. \tag{6.126}$$

Так как $\hat{E}_0(\nu)$ — выпуклая вверх функция, то $R = \hat{E}_0(\nu)/\nu \geqslant \hat{E}_0'(\nu)$ и, следовательно,

$$\eta = \frac{\nu R}{\hat{E}_0(\nu) - \nu \hat{E}_0'(\nu)} - 1 \geqslant 0, \tag{6.127}$$

т. е. η — неотрицательная величина. Из формул (6.117), (6.123) и (6.126) получаем

$$\min_{0 \leqslant \eta \leqslant \infty} (\eta + 1) E_{spL}(R, \eta) = \nu R = \hat{E}_0(\nu). \tag{6.128}$$

Далее, предположим, что η кратно $1/(m+1)$, введем «корректирующее слагаемое» $\delta < 1/(m+1)$ и исследуем его влияние. Изменение η на величину порядка $1/(m+1)$ приводит к изменению показателя экспоненты на величину порядка $n_A/(m+1) = n_0$. Обозначим n_0/n_A через $O(n_A)$. Величина R связана с n следующим образом:

$$R = \frac{\eta + 1 + \frac{1}{m+1}}{\eta + \frac{1}{m+1}} \hat{E}_0'(\nu) = \frac{(\eta+1)\left(1 + \frac{1}{(m+1)(\eta+1)}\right)}{\eta\left(1 + \frac{1}{(m+1)\eta}\right)} \hat{E}_0'(\nu) \simeq$$

$$\simeq \frac{\eta+1}{\eta} \hat{E}_0'(\nu)\left\{1 + \frac{1}{(m+1)(\eta+1)}\right\}\left(1 - \frac{1}{(m+1)\eta}\right) \simeq$$

$$\simeq \frac{\hat{E}_0(\nu)}{\nu}\left\{1 - \frac{1}{(m+1)\eta(\eta+1)}\right\}.$$

Чтобы учесть влияние δ, в показатель экспоненты оценки для p_E достаточно ввести слагаемое $O(n_A)$. Из формул (6.119), (6.120), (6.126) и (6.128) получаем следующую теорему.

Теорема 6.5. (Витерби.) *В дискретном канале без памяти вероятность ошибки при декодировании произвольного сверточного кода с кодовым ограничением n_A ограничена снизу следующим образом:*

$$p_E > \exp\{-n_A[E_{spL}(R) + O(n_A)]\}, \tag{6.129}$$

где

$$E_{spL}(R) = \hat{E}_0(\nu), \qquad 0 \leqslant \nu < \infty, \tag{6.130}$$

$$R = \hat{E}_0(\nu)/\nu. \tag{6.131}$$

В случае малых скоростей, используя прямолинейную границу, для p_E можно получить более сильную нижнюю границу. Эту границу мы приведем здесь без доказательства.

Утверждение 6.1. *В области малых скоростей имеет место следующая граница:*

$$p_E > \exp\{-n_A[E_{spL}(R) + O(N)]\},$$
$$E_{spL}(R) = E_u, \quad 0 \leqslant R \leqslant \frac{E_0(\tilde{\nu})}{\tilde{\nu}}, \tag{6.132}$$

$$E_u = \max_{Q(u)} \left\{ -\lim_{\nu\to\infty} \left[\nu \ln \sum_u \sum_{u'} Q(u) Q(u') \times \right.\right.$$
$$\left.\left. \times \left(\sum_v \sqrt{p(v \mid u)\, p(v \mid u')} \right) \right]^{1/\nu} \right\}, \tag{6.133}$$

где $\tilde{\nu}$ — решение уравнения $\hat{E}_0(\tilde{\nu}) = E_\infty$ и

$$E_\infty = \max_{Q(u)} -\sum_u \sum_{u'} Q(u) Q(u') \ln \sum_v \sqrt{p(v \mid u)\, p(v \mid u')}. \tag{6.134}$$

6.7. Гибридные методы кодирования

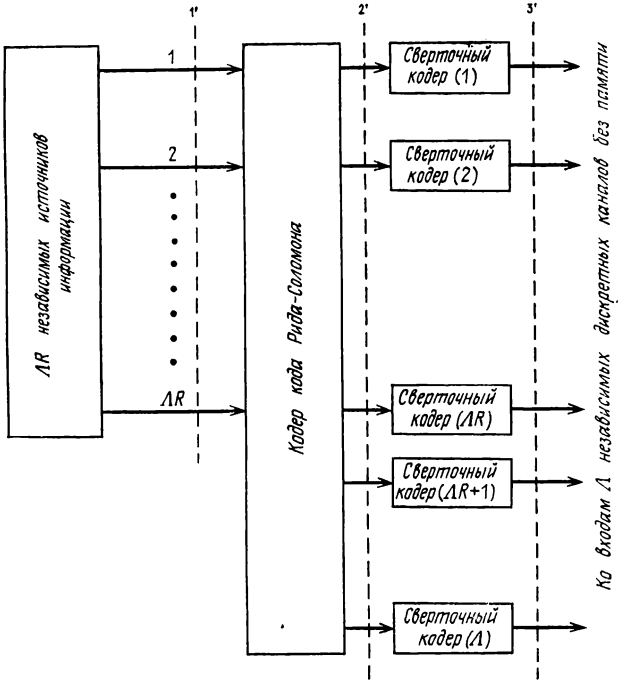
Как было указано выше, важнейшей характеристикой алгоритмов последовательного декодирования является вероятность переполнения буфера. Вероятность переполнения буфера можно понизить, если установить для последовательного декодера максимально допустимое число операций, а символы, для определения которых декодер должен выполнить большее число операций, декодировать другим методом, например одним из алгебраических методов, для которых число операций при декодировании всегда ограничено. Возможность уменьшения вероятности переполнения буфера указанным выше образом строго обосновал Фалконер [16], воспользовавшись для этого рассматриваемым ниже гибридным методом кодирования и декодирования.

6.7.1. Системы гибридного кодирования

Предположим, что по Λ независимым дискретным каналам без памяти каждые ν секунд передается один символ канала (фиг. 6.14). Будем считать, что на вход каждого из этих каналов символы подаются с выхода относящегося к этому каналу сверточного кодера. Если используемые в этой системе сверточные коды имеют скорость передачи $r = 1/\varpi$, то на вход любого сверточного кодера информационные двоичные символы должны поступать каждые $\varpi\nu$ секунд. Наконец, предположим, что через каждые $\lambda\varpi\nu$ секунд регистры сдвига всех сверточных

кодеров приводятся в нулевое состояние, так что очередные информационные символы вводятся в регистры, содержащие лишь нулевые символы. В этом случае каждый блок из λ входных двоичных символов кодируется в блок из λw символов канала независимо от входных двоичных символов предшествующих и последующих блоков.

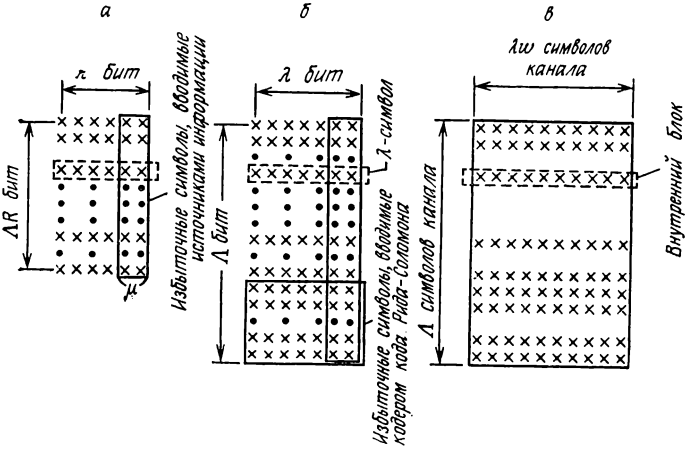
Чтобы символы, поступающие на входы кодеров, при декодировании можно было восстановить с малой вероятностью



Фиг. 6.14. Гибридный кодер.

ошибки, как следует из формулы (6.95), они должны оказывать влияние на достаточно большое число входных символов канала или, другими словами, кодовое ограничение используемых сверточных кодов должно быть достаточно большим. Однако поскольку перед передачей каждого нового блока из входных символов регистры сдвига кодеров очищаются, то последние символы каждого блока оказывают воздействие уже на сравнительно небольшое число символов канала. Это приводит к тому, что вероятность ошибки при декодировании последних символов каждого блока возрастает. Для того чтобы предотвра-

тить возрастание вероятности ошибки в конце блока, в качестве последних $\mu(\mu < \lambda)$ входных символов в каждом блоке берется некоторая фиксированная последовательность, известная декодеру. В этом случае входные символы каждого кодера, которые декодеру неизвестны априори, оказывают влияние по крайней мере на μw символов канала. Однако в результате последние μ символов каждого входного блока оказываются избыточными и реальная скорость передачи сверточных кодеров становится равной $(1 - \mu/\lambda)/w$. Вообще говоря, если выбрать $\lambda \gg \mu$, то этим уменьшением скорости передачи можно пренебречь.



Фиг. 6.15. Структура кода (фигуры а, б, в соответствуют сечениям 1′, 2′ и 3′ на фиг. 6.14).

а — входные символы кодера кода Рида — Соломона; *б* — параллельный блок; *в* — суперблок.

а) *Структура кодов.* Структура кодового слова при гибридном кодировании показана на фиг. 6.15. Как видно из фиг. 6.15, *в*, Λ внутренних блоков формируются при кодировании одновременно и передаются по Λ дискретным каналам без памяти. Эти Λ внутренних блоков образуют один суперблок. Как указано на фиг. 6.15, *в*, $\Lambda\lambda$-символов, поступающих параллельно на входы последовательных декодеров, образуют параллельный блок. Верхние ΛR λ-символов параллельного блока составляют независимые подблоки, каждый из которых состоит из $\lambda - \mu$ информационных двоичных символов и μ известных фиксированных символов. Остальные $\Lambda(1 - R)$ λ-символов параллельного блока являются проверочными и формируются алгебраическим кодером по ΛR верхним независимым λ-символам, с которыми алгебраический кодер оперирует как с элементами поля $GF(2^{\lambda})$.

Параллельные блоки образуют 2^{λ}-ичный блоковый код длины Λ, содержащий $2^{\lambda\Lambda R}$ кодовых слов.

Скорость передачи для всей описанной выше системы кодирования, выраженная в битах на один используемый символ канала, равна

$$R' = R(1 - \mu/\lambda)/w \text{ [бит]}.$$

Так как по предположению по каждому дискретному каналу без памяти передается один символ канала каждые ν секунд, то скорость передачи информации равна $\Lambda R'/\nu$ [бит/с].

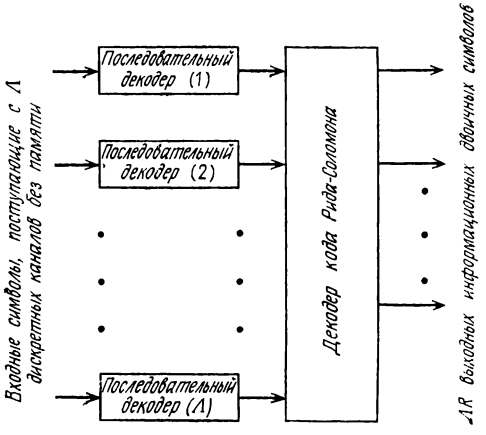
Далее остановимся на выборе блокового кода. При средних и больших длинах блоков наиболее эффективными кодами являются коды Рида — Соломона. Эти коды могут быть построены для любой скорости передачи R и любой длины Λ, не превышающей объема алфавита символов, и имеют минимальное расстояние $d = d_{\text{макс}} = (1 - R)\Lambda + 1$. Если рассматривать коды над полем $GF(2)$ или его расширением, то размер алфавита является степенью числа 2 и должен быть не меньше $\Lambda + 1$. Последнее условие эквивалентно следующему: $\lambda \geqslant \log(\Lambda + 1)$.

Для построения кодера Рида — Соломона требуется число элементов, линейно растущее с Λ. Каждый из Λ сверточных кодеров содержит самое большее ν разрядов регистра сдвига. Следовательно, общее число элементов, необходимых для построения гибридного кодирующего устройства, пропорционально $\nu\Lambda$.

б) *Декодер.* Как показано на фиг. 6.16, гибридный декодер состоит из Λ параллельно работающих последовательных декодеров и декодера кода Рида — Соломона. При декодировании каждого суперблока Λ независимых последовательных декодеров Фано осуществляют декодирование соответствующих внутренних блоков. Декодирование различных внутренних блоков, вообще говоря, может продолжаться различное время. Здесь будем предполагать, что после окончания декодирования последовательными декодерами некоторых $\Lambda - D$ внутренних блоков декодирование остальных D блоков прекращается и все Λ последовательных декодеров начинают декодирование внутренних блоков следующего суперблока. При этом λ-символы, декодирование которых не было завершено, воспринимаются далее декодером кода Рида — Соломона как стирания.

Таким образом, после завершения декодирования суперблока последовательными декодерами он переходит в параллельный блок (параллельный блок — это кодовое слово Рида — Соломона, быть может, искаженное в канале), состоящий из $(\Lambda - D)$ λ-символов, последовательное декодирование которых было завершено, и D λ-символов, которые рассматриваются как

стирания. Этот параллельный блок подается на декодер кода Рида — Соломона. Предположим, что используемый код Рида — Соломона имеет минимальное расстояние d и что среди $\Lambda - D$ λ-символов, последовательное декодирование которых было завершено, имеется не более чем T неправильно декодированных λ-символов. Как хорошо известно, если выполняется соотношение $2T + D = d - 1$ между числом ошибок T, числом стираний D и минимальным расстоянием кода d, то декодер кода Рида — Соломона может декодировать данный блок правильно. Итак, при гибридном кодировании и декодировании число операций при декодировании всегда конечно и приближенно оценивается величиной порядка $\Lambda \log \Lambda$ или $T \log \Lambda$. Как



Фиг. 6.16. Гибридный декодер.

было указано, если даже последовательное декодирование некоторых D внутренних блоков не завершено, но другие $\Lambda - D$ внутренних блоков уже декодированы, то начинается процесс коррекции всех этих блоков с помощью алгебраического декодера. Таким образом, использование алгебраического декодера позволяет не допустить резких возрастаний объема вычислений, время от времени возникающих при последовательном декодировании внутренних блоков, и таким образом уменьшить вероятность переполнения буферов последовательных декодеров.

В заключение заметим, что так как минимальное расстояние кода Рида — Соломона равно $(1 - R)\Lambda + 1$, то всегда можно положить $2T + D = (1 - R)\Lambda$, т. е. число исправляемых ошибок и стираний при фиксированной скорости передачи R прямо пропорционально Λ.

6.7.2. Характеристики гибридного кодирования

Найдем вероятность ошибки, распределение числа операций, среднее число операций и вероятность переполнения буфера для описанного выше гибридного метода кодирования. Сначала рассмотрим случай, когда половина минимального расстояния кода Рида — Соломона используется для исправления стираний, а другая — для исправления ошибок. Будем использовать код Рида — Соломона для коррекции $T = (\Lambda\varepsilon/2) - 1$ ошибок и $D = \Lambda\varepsilon - 1$ стираний, где ε — некоторое число из интервала $0 < \varepsilon < 1/2$. Подставляя эти значения T и D в равенство $2T + D = (1 - R)\Lambda$, находим ε:

$$\varepsilon = \frac{1-R}{2} + \frac{3}{2\Lambda} \geqslant \frac{1-R}{2}. \tag{6.135}$$

Заметим, что при больших значениях Λ параметр ε не зависит от Λ. Полагая $\mu/\lambda = \varepsilon$, получаем

$$R' = rR(1 - \mu/\lambda) = r(1 - \varepsilon)\left(1 - 2\varepsilon + \frac{3}{\Lambda}\right) > r(1 - \varepsilon)(1 - 2\varepsilon). \tag{6.136}$$

а) *Вероятность ошибки.* Начнем с отыскания вероятности ошибки. Согласно результатам Юдкина [13], связанным с исследованием вероятности ошибки для алгоритма Фано, вероятность p_{Eu} ошибки при последовательном декодировании одного внутреннего блока можно оценить сверху экспоненциальной функцией, стремящейся к нулю с ростом числа избыточных двоичных символов, содержащихся в каждом λ-символе, поступающем на вход кодера. Используя то, что $\mu = \lambda\varepsilon$, получаем

$$p_{Eu} < p'_{Eu} = \lambda G_e \exp[-\lambda\varepsilon E_u(r)], \tag{6.137}$$

где G_e — постоянная, а $E_u(r)$ — функция скорости передачи r древовидного кода и переходных вероятностей дискретного канала без памяти. Вероятность p_E ошибки для гибридного декодера равна вероятности того, что $\Lambda\varepsilon/2$ или более внутренних блоков декодированы неверно; она определяется равенством

$$p_E = \sum_{v=\lambda\varepsilon/2}^{\Lambda} \binom{\Lambda}{v} p_{Eu}^{v}(1 - p_{Eu})^{\Lambda - v}. \tag{6.138}$$

Правая часть (6.138) представляет собой функцию распределения суммы независимых двоичных случайных величин, и ее можно оценить сверху с помощью неравенства Чернова (см., на-

пример, Возенкрафт и Джекобс [24]) [1]). Используя это неравенство, находим

$$p_E \leqslant \exp(-\Lambda\{T_e[\varepsilon/2,\ p_{Eu}] - H(\varepsilon/2)\}), \qquad 0 \leqslant p_{Eu} \leqslant \varepsilon/2, \tag{6.139}$$

где

$$T_e(u,\ v) = -u\ln v - (1-u)\ln(1-v),$$
$$H(u) = -u\ln u - (1-u)\ln(1-u).$$

Как нетрудно проверить, при $v < u < 1/2$

$$T_e(u,\ v) - H(u) > 0. \tag{6.140}$$

Отсюда следует, что вероятность ошибки p_E с ростом Λ экспоненциально стремится к нулю. Так как при $p_{Eu} < \varepsilon/2$

$$\frac{\partial}{\partial p_{Eu}} T_e[\varepsilon/2,\ p_{Eu}] < 0, \qquad p_{Eu} < \varepsilon/2, \tag{6.141}$$

то показатель экспоненты в формуле (6.139) является монотонно убывающей функцией p_{Eu}. Следовательно, в (6.139) вместо p_{Eu} можно подставить верхнюю границу p'_{Eu}, задаваемую формулой (6.137). В результате для p_E получаем следующую верхнюю границу:

$$p_E < \exp[-\Lambda\{T_e(\varepsilon/2,\ p'_{Eu}) - H(\varepsilon/2)\}], \qquad p'_{Eu} < \varepsilon/2. \tag{6.142}$$

б) *Распределение числа операций.* Далее найдем распределение числа операций. Пусть c — число операций, которое необходимо последовательному декодеру для того, чтобы без помощи алгебраического декодера декодировать один внутренний блок. Как следует из разд. 6.4, эта величина имеет распределение Парето

$$p(c \geqslant L) < GL^{-r_{\text{выч}}/r}, \tag{6.143}$$

где G — некоторая постоянная. Этим неравенством можно воспользоваться для того, чтобы получить верхнюю границу для числа операций, необходимых гибридному декодеру для декодирования одного суперблока.

Чтобы упростить последующий анализ, предположим, что последовательные декодеры всех внутренних блоков не начинают декодирование очередного суперблока до тех пор, пока

1) не закончат декодирование предыдущего суперблока;

2) в буферы последовательных декодеров не поступит после окончания приема весь суперблок.

Эти предположения означают, что все суперблоки декодируются независимо и, кроме того, в процессе декодирования

[1]) В частном случае неравенство Чернова выводится также в разд. 6.5 данной книги.

последовательные декодеры не простаивают в ожидании поступления новых ребер. При этих предположениях последовательные декодеры работают несколько медленнее, чем обычно, и полученные ниже значения вероятности переполнения буфера оказываются завышенными.

Итак, при декодировании каждого суперблока сначала с помощью последовательных декодеров декодируются $\Lambda - D$ из Λ внутренних блоков, после чего параллельный блок декодируется декодером кода Рида — Соломона. Так как число операций при декодировании, выполняемых декодером кода Рида — Соломона, конечно, то здесь это число мы учитывать не будем. Сначала введем понятие приведенного числа операций c, необходимого для декодирования одного суперблока. Пусть c_j — число операций, необходимых j-му последовательному декодеру для декодирования j-го внутреннего блока. Если величины $c_1, \ldots, c_\Lambda$ упорядочить в порядке их убывания, то по определению c — это $(D+1)$-е из этих чисел. Определив таким образом c, можно утверждать, что при декодировании любого суперблока ни один из последовательных декодеров не выполняет более чем c операций. Если $D+1$ или больше чисел $c_1, c_2, \ldots$ $\ldots, c$ превышают L, то приведенное число операций c также будет превышать L. Как следует из неравенства (6.143),

$$p[c_j \geqslant L] \leqslant p_l = G \cdot L^{-\gamma}, \tag{6.144}$$

где $\gamma = r_{\text{выч}}/r$. Точно так же, как при выводе неравенства (6.142), полагая $D+1 = \Lambda\varepsilon$, из формулы (6.138) получаем

$$p[c \geqslant L] \leqslant \exp[-\Lambda\{T_e(\varepsilon, p_e) - H(\varepsilon)\}], \qquad p_e < \varepsilon. \tag{6.145}$$

Более простую границу можно получить, если воспользоваться тем, что $T_e(\varepsilon, p_e) \geqslant -\varepsilon \ln p_e$. Действительно, используя это неравенство, находим, что при всех $p_e < \varepsilon$

$$p[c \geqslant L] \leqslant \exp[\Lambda H(\varepsilon)]\, p^{\Lambda\varepsilon} = (G_1/L)^{\Lambda\varepsilon\gamma}, \tag{6.146}$$

где

$$G_1 = \exp[H(\varepsilon)/\gamma\varepsilon]\, G^{1/\gamma}. \tag{6.147}$$

Как следует из определений G_1 и $H(\varepsilon)$, а также определения p_l [см. формулу (6.144)], при $L > G_1$ условие $p_l < \varepsilon$ всегда выполняется. Учитывая, что величина $p[c \geqslant L]$ является вероятностью и поэтому не больше 1, получаем

$$p[c \geqslant L] \leqslant \begin{cases} [G_1/L]^{\Lambda\varepsilon\gamma}, & L > G_1, \\ 1, & L \leqslant G_1. \end{cases} \tag{6.148}$$

Используя это неравенство, оценим среднее число операций, необходимое для декодирования одного суперблока при условии,

что $0 \leqslant \gamma \leqslant 1$ и $\Lambda\varepsilon\gamma > 1$. Сначала найдем среднее значение приведенного числа операций c:

$$\langle c\rangle_{\text{ср}} = \sum_{L=1}^{\infty} Lp[c=L] = \sum_{L=1}^{\infty} L\{p[c \geqslant L] - p[c \geqslant L+1]\} = \sum_{L=1}^{\infty} p[c \geqslant L].$$

Отсюда и из формулы (6.148) следует, что

$$\langle c\rangle_{\text{ср}} \leqslant G_1 + \sum_{L=G_1+1}^{\infty} (G_1/L)^{\Lambda\varepsilon\gamma}.$$

Так как $(G_1/L)^{\Lambda\varepsilon\gamma}$ является положительной монотонно убывающей функцией L, то сумму в правой части последнего неравенства можно заменить интегралом по L от G_1 до ∞:

$$\langle c\rangle_{\text{ср}} \leqslant G_1 + \int_{G_1}^{\infty} (G_1/L)^{\Lambda\varepsilon\gamma}\, dL = \frac{\Lambda\varepsilon\gamma G_1}{\Lambda\varepsilon\gamma - 1} = \frac{\Lambda\varepsilon\gamma}{\Lambda\varepsilon\gamma - 1}\left[\exp\{H(\varepsilon)/\gamma\varepsilon\}\, G^{1/\gamma}\right]. \quad (6.149)$$

При $\Lambda\varepsilon\gamma > 1$ правая часть (6.149) положительна и дает оценку сверху для среднего приведенного числа операций, необходимого для декодирования одного суперблока. Заметим, что для любой скорости передачи, сколь угодно близкой к пропускной способности канала, при достаточно больших Λ и выполнении неравенства $\Lambda\varepsilon\gamma > 1$ приведенное число операций на один суперблок является ограниченным.

в) *Вероятность переполнения буфера.* Остановимся на вероятности переполнения буфера при гибридном декодировании. Каждые $\lambda w\nu$ секунд в декодер поступает один суперблок.

Каждый из последовательных декодеров имеет буфер, который он использует для хранения последних Bw символов, поступивших с выхода соответствующего дискретного канала без памяти. Так как ранее предполагалось, что перед началом декодирования очередного суперблока все символы последнего должны быть приняты, то буферы должны иметь объем, достаточный для хранения по крайней мере одного суперблока. Следовательно, параметр B должен быть не меньше λ. Часть суперблоков, хранящихся в буфере и еще не декодированных, образует очередь.

Если очередь превышает B/λ суперблоков (или Bw символов канала на каждый дискретный канал без памяти), то возникает переполнение буфера. Интересующей нас величиной является $p_L(B)$-вероятность возникновения переполнения буфера за время декодирования L первых суперблоков при условии, что

декодер начинает декодирование очередного суперблока, имея свободный буфер. Верхняя граница для вероятности $p_L(B)$ была получена Фалконером. Эта граница имеет вид

$$p_L(B) < (1 + e^6) L \left[\frac{G_1 e}{\sigma(B - \lambda)}\right]^{\Lambda \varepsilon \gamma}, \qquad (6.150)$$

где σ — некоторая постоянная, зависящая от скорости вычислений.

г) *Численный пример.* В заключение приведем численный пример, иллюстрирующий характеристики гибридного декодера.

Фалконер и Ниссен [26] провели моделирование последовательного декодирования, используя сверточный код со скоростью $r = 1/7$ и осуществляя в канале с отношением сигнал/шум, равным —6,5 дБ, квантование сигнала на восемь уровней. При этом $\gamma \simeq 1$ и, следовательно, $r_{\text{выч}} \simeq 1/7$. Будем считать, что длина внутреннего блока ν равна 360 ребрам, из которых $m = 24$ ребра являются избыточными. Если кодовое ограничение сверточного кода выбрать равным 24 ребрам, то истинная скорость передачи будет равна

$$\frac{1}{7} \cdot \frac{360 - 24}{360} = 0{,}133 \text{ бит на одно использование канала.} \qquad (6.151)$$

Код Рида — Соломона выберем длины $\Lambda = 31 = 2^5 - 1$ со скоростью $R = 26/31$ и алфавитом из $2^5 = 32$ символов. В этом случае каждый суперблок будет состоять из 72 кодовых слов кода Рида — Соломона. Из 31 внутреннего блока каждого суперблока пять внутренних блоков являются проверочными. Предположим, что код Рида — Соломона используется только для исправления стираний.

Фалконер и Ниссен экспериментально исследовали распределение суммарного числа операций при декодировании одного внутреннего блока. Согласно полученным ими данными, для канала, отношение сигнал/шум в котором равно —6,5 дБ, вероятность того, что суммарное число операций c при декодировании одного суперблока превысит 36 000, приближенно равна 10^{-2}. Следовательно, вероятность Pr ($c \geqslant 36\,000$) того, что приведенное число операций, необходимое для декодирования одного суперблока, превысит 36 000, равна вероятности того, что среди 31 внутреннего блока шесть или более блоков требуют для своего декодирования более 36 000 операций. Эта вероятность находится по формуле

$$P_r(c \geqslant 36\,000) = \sum_{v=6}^{31} \binom{31}{v} (10^{-2})^v (1 - 10^{-2})^{31-v} \simeq 6 \cdot 10^{-7}. \qquad (6.152)$$

Если для каждого последовательного декодера $\sigma = 50$, то за время ввода в буфер нового внутреннего блока декодер может

выполнить $360\,\sigma = 18\,000$ операций. Следовательно, если каждый последовательный декодер имеет буфер, способный хранить три внутренних блока, то переполнение буфера будет возникать только в том случае, если приведенное число операций, необходимых для декодирования первого суперблока, превышает $2 \times 18\,000 = 36\,000$. Если допустить, что переполнения буфера возникают достаточно редко и являются статистически независимыми, то вероятность переполнения буфера при декодировании одного суперблока будет величиной порядка $6 \cdot 10^{-7}$. Так как буфер каждого последовательного декодера позволяет хранить 3×360 ребер, то общее число ребер, которые могут храниться в памяти декодера, равно $1080\,\Lambda = 33\,480$. Общее число двоичных символов, входящих в один суперблок, равно $360 \times 31 = 11\,160$.

Полагая $\lambda\varepsilon = m = 24$, $w = 7$, $E_u(r)\log_2 \approx R_{\text{выч}} \approx 1/7$ и в (6.137) $G_e \approx 1$, для вероятности ошибочного декодирования одного внутреннего блока получаем следующую границу:

$$p_{Eu} \leqslant 360 \times 2^{-24} = 2{,}23 \cdot 10^{-5}. \tag{6.153}$$

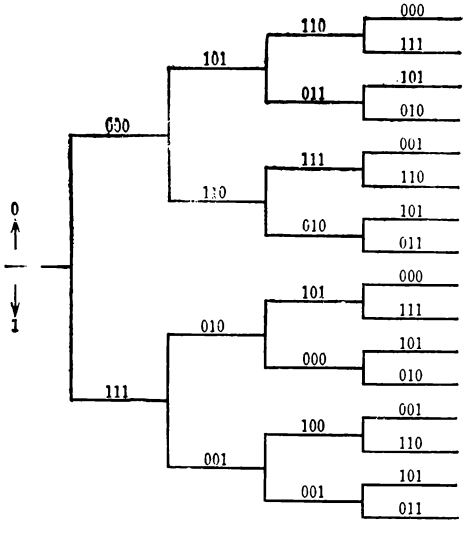
С помощью несложных расчетов, основанных на результатах моделирования, можно оптимизировать различные параметры гибридных систем кодирования и при этом учесть требования к стоимости и другим характеристикам системы в целом.

6.8. Стек-алгоритм

Выше было проведено детальное исследование алгоритма Фано. Этот алгоритм обладает рядом несомненных достоинств, но, как можно было убедиться выше, анализ многих его характеристик очень сложен. Рассматриваемый в этом разделе так называемый стек-алгоритм, найденный независимо Джелинеком [7] и Зигангировым [8], по сравнению с алгоритмом Фано может быть описан гораздо проще, а кроме того, он достаточно прост для понимания. Этот алгоритм не всегда лучше алгоритма Фано (Гейст [15]), но благодаря своей простоте он позволяет глубже понять сущность последовательного декодирования. Для простоты рассмотрим здесь этот алгоритм применительно к двоичным кодам.

Для пояснений будем использовать двоичный древовидный код со скоростью $R = 1/3$, кодовое дерево которого изображено на фиг. 6.17. Пусть i_t — двоичный информационный символ, поступающий в момент времени t на вход кодера. Пути длины t будем обозначать с помощью соответствующих им векторов $\mathbf{i}^t = (i_1, i_2, \ldots, i_t)$. Через $u_j(\mathbf{i}^t)$, $j < l$, обозначим символы, используемые для передачи по каналу j-го ребра пути $\mathbf{i}^t$. Так как

передаваемые символы, соответствующие ребрам показанного на фиг. 6.17 кодового дерева, представляют собой последовательности из трех двоичных символов, то для них удобно использовать восьмеричное представление, например $u_3(0\,1\,0\,i_4) = 7$, $u_2(1, 0, i_3, i_4) = 2$. Предположим, что на выходе канала принята последовательность v^p, где p — длина этой последовательности в ребрах кодового дерева. Задача состоит в том, чтобы по этой последовательности восстановить входную последовательность i^p.



Фиг. 6.17. Пример древовидного кода со скоростью $R = 1/3$.

Предположим, что символ канала u, $u = 0, 1, \ldots, 7$, поступает на вход канала с вероятностью $p(u)$. Тогда вероятность $p(v)$ появления на выходе канала символа v будет задаваться равенством

$$p(v) = \sum_u p(v \mid u)\, p(u). \tag{6.154}$$

При декодировании принятой последовательности v^p находятся функции правдоподобия $L(i^t)$ для различных путей $i^t = (i^s, i_{s+1}, \ldots, i_t)$:

$$L(i^t) = \sum_{s=1}^{t} \lambda_s(i^s). \tag{6.155}$$

Для этого используются функции правдоподобия $\lambda_s(i^s)$ ребер, соединяющих узлы i^{s-1} и i^s:

$$\lambda_s(i^s) = \log_2\left[\frac{p\,[v_s \mid u_s(i^s)]}{Q(v_s)}\right] - n_0 R. \tag{6.156}$$

Поскольку в процессе декодирования последовательность v^n не изменяется и символы $u_s(i^s)$ для каждого i^s определяются кодовым деревом, которое считается заданным, то величины $L(i^t)$ и $\lambda_s(i^s)$ являются функциями только путей. Рассмотрим следующий алгоритм декодирования:

1. Вычисляются функции правдоподобия $L(0)=\lambda_1(0)$ и $L(1)=\lambda_1(1)$ двух ребер, выходящих из начального узла кодового дерева. Эти значения помещаются в память декодера.

2. Если $L(0)\geqslant L(1)$, то $L(0)$ извлекается из памяти декодера и вычисляются значения $L(00)=L(0)+\lambda_2(00)$ и $L(01)=$ $=L(0)+\lambda_2(01)$. Если $L(0)<L(1)$, то из памяти извлекается $L(1)$ и вычисляются значения $L(10)=L(1)+\lambda_2(10)$ и $L(11)=$ $=L(1)+\lambda_2(11)$. Величина $L(1)$ ($L(0)$) удаляется из памяти декодера, а вновь найденные значения функций правдоподобия помещаются в память декодера. На этом шаг 2 заканчивается. Таким образом, перед следующим шагом в памяти декодера хранятся значения функций правдоподобия трех путей, а именно двух путей длиной 2 и одного пути длиной 1.

3. Значения функций правдоподобия трех путей, хранящиеся в памяти декодера, упорядочиваются в порядке их убывания, после чего находится путь, соответствующий наибольшему из этих значений. Для двух путей, которые получаются из указанного выше пути присоединением следующего ребра, вычисляются значения функции правдоподобия. Значение функции правдоподобия пути, который был только что удлинен, из памяти удаляется, а в память помещаются вновь вычисленные значения функций правдоподобия. Например, предположим, что в памяти декодера хранятся значения $L(0)$, $L(10)$ и $L(11)$ и $L(10)\geqslant L(0)\geqslant L(11)$. В этом случае из памяти удаляется $L(0)$, а помещаются в память значения функций правдоподобия $L(100)=L(10)+\lambda_3(100)$ и $L(101)=L(10)+\lambda_3(101)$. На этом шаг декодирования заканчивается. Перед следующим шагом декодирования в памяти декодера хранятся значения функций правдоподобия четырех путей, среди которых два пути имеют длину 3, один — длину 1, и еще один — длину 2 (возможно, что все четыре пути будут иметь длину 2).

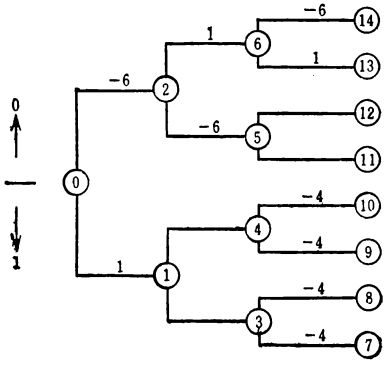
4. Каждый последующий шаг алгоритма декодирования аналогичен описанным выше шагам. После k-го шага в памяти декодера хранятся $k+1$ значений функций правдоподобия $k+1$ путей, которые могут иметь различную длину, но ни один из которых не является началом другого. На $(k+1)$-м шаге находятся наибольшее содержащееся в памяти значение функции правдоподобия и соответствующий этому значению путь. Значение функции правдоподобия этого пути из памяти декодера удаляется, а значения функций правдоподобия двух путей, полу-

чающихся из указанного выше пути присоединением следующего ребра, помещаются в память.

5. Процесс декодирования заканчивается, когда длина пути, который должен быть удлинен, оказывается равной некоторому заданному числу p.

Так как содержимое памяти декодера постоянно упорядочивается, то этот алгоритм получил название стек-алгоритма.

Поясним этот алгоритм на примере. Рассмотрим кодовое дерево, узлы которого пронумерованы так, как показано на фиг. 6.18. Для обозначения путей будем использовать номера



Фиг. 6.18. Пример распределения значений функций правдоподобия по различным путям.

узлов, в которых они оканчиваются. Числа, приведенные над каждым ребром, являются значениями функции правдоподобия соответствующего ребра для некоторой принятой последовательности v^3. Так, значение функции правдоподобия пути с номером 13 равно $-6+1+1=-4$. В рассматриваемом примере состояние памяти декодера меняется следующим образом (номера путей в каждой строке упорядочены таким образом, что значение функции правдоподобия пути, номер которого занимает крайнее левое положение, максимально):

Состояние	Путь
1	0
2	1, 2
3	4, 3, 2
4	3, 2, 10, 9
5	2, 10, 9, 8, 7
6	6, 10, 9, 8, 7, 5
7	13, 10, 9, 8, 7, 14, 5

Так как путь с номером 13, занимающим крайнее левое положение в последней строке, имеет длину 3, то декодер на шаге 7 декодирование прекращает.

Таким образом, стек-алгоритм действительно значительно проще алгоритма Фано. В работе [7] приведено обобщение этого алгоритма на случай недвоичных кодов. Анализ вероятности ошибки и среднего числа операций при декодировании можно найти в работе [25]. Методы расчета этих характеристик для стек-алгоритма аналогичны методам расчета соответствующих характеристик алгоритма Фано и здесь рассматриваться не будут [1]).

6.9. Структура расстояний сверточных кодов

Исследование расстояний между кодовыми словами является важным при анализе свойств кодов. Следуя Месси [17, 18], получим сначала аналоги верхней границы Плоткина и нижней границы Варшамова — Гилберта для минимального расстояния систематических сверточных кодов. Далее остановимся на введенном Костелло [19] понятии свободного расстояния, которым удобно пользоваться для описания расстояния кодов, используемых при последовательном декодировании, и в заключение рассмотрим несколько алгоритмов построения сверточных кодов со скоростью $1/n_0$.

Как уже указывалось выше, существуют два типа методов алгебраического декодирования сверточных кодов: декодирование с обратной связью и дефинитное декодирование. При выводе кодовых границ для минимального расстояния сверточных кодов, допускающих декодирование одним из этих методов, коды удобно задавать с помощью порождающих многочленов, как это было описано в разд. 6.2. Если совокупности проверочных и информационных символов на выходе систематического сверточного кодера в момент времени u обозначить соответственно через

$$\boldsymbol{t}_u^* = \left(t_u^{(k_0+1)}, t_u^{(k_0+2)}, \ldots, t_u^{(n_0)}\right) \tag{6.157}$$

и

$$\boldsymbol{i}_u = \left(i_u^{(1)}, i_u^{(2)}, \ldots, i_u^{(k_0)}\right), \tag{6.158}$$

то систематический сверточный код можно задать с помощью порождающих многочленов следующим образом:

$$\boldsymbol{t}_u^* = \boldsymbol{i}_u G_0'' + \boldsymbol{i}_{u-1} G_1'' + \ldots + \boldsymbol{i}_0 G_u''. \tag{6.159}$$

При рассмотрении декодирования с обратной связью всегда будем считать, что $\boldsymbol{i}_u = 0$ при $u > 0$.

[1]) См. также книгу К. Ш. Зигангирова «Процедуры последовательного декодирования», изд-во «Связь», М., 1974. — *Прим. ред.*

6.9.1. Верхняя граница Плоткина при декодировании с обратной связью

Как указывалось выше, при декодировании с обратной связью без ограничения общности всегда можно рассматривать лишь декодирование подблока i_0 (декодирование остальных подблоков i_u осуществляется аналогично). Минимальное расстояние сверточного кода d_{FD} при декодировании с обратной связью естественно определить следующим образом.

Минимальное расстояние d_{FD} сверточного кода есть наименьшее число символов, в которых различаются две кодовые последовательности длиной $n_A = n_{FD}$, порождаемые кодером с момента времени 0 до момента времени m и имеющие отличные первые информационные подблоки i_0.

В силу свойства линейности систематических сверточных кодов определенное таким образом минимальное расстояние совпадает с минимальным весом начальных кодовых слов:

$$d_{FD} = \min_{i_1 \neq 0} W_H(i_0, i_1, \ldots, i_m, t_0^*, t_1^*, \ldots, t_m^*), \qquad (6.160)$$

где $W_H(\cdot)$ — вес Хэмминга вектора, заключенного в круглые скобки. В целях упрощения далее будем рассматривать лишь двоичные коды.

Из равенства (6.160) непосредственно следует, что максимальное значение d_{FD} в классе всех сверточных кодов со скоростью $R = k_0/n_0$, порождаемых кодерами с m-разрядными регистрами сдвига, является верхней границей для d_{FD}. Это максимальное значение обозначим через $\max_{FD}(m, n_0, k_0)$. Для $\max_{FD}(m, n_0, k_0)$ Месси была получена следующая верхняя граница:

$$\max_{FD}(m, n_0, k_0) \leqslant \left[\frac{m+5}{2}\right](n_0 - k_0) + 1. \qquad (6.161)$$

Для доказательства этого неравенства нам понадобятся следующие свойства величины $\max_{FD}(m, n_0, k_0)$ как функции m, n_0 и k:

Свойство 1.

$$\max_{FD}(m, n_0, k_0) \leqslant \max_{FD}(m, n_0 - k_0 + 1, 1). \qquad (6.162)$$

Так как минимум в (6.160) берется по всем кодовым последовательностям, каждый подблок из n_0 символов которых содержит k_0 информационных символов, то естественно, что минимум в (6.162) берется и по всем кодовым последовательностям, порождаемым информационными подблоками i_u, последние $k_0 - 1$ символов которых равны нулю. Минимизация только по таким кодовым последовательностям эквивалентна нахождению минимального расстояния сверточного кода, на вход кодера которого в каждый момент времени подается один информационный сим-

вол, на выходе кодера формируется $n_0-(k_0-1)$ символов и порождающие матрицы G_j'', $j=0, 1, \ldots, u$, которого представляют собой первые строки соответствующих порождающих матриц исходного кода. Максимум минимальных расстояний по всем таким кодам равен правой части (6.162). Это доказывает справедливость неравенства (6.162).

Свойство 2. Для любого нечетного n_0

$$\max_{FD}(m, n_0, 1) \leqslant n_0 + m\frac{n_0-1}{2}. \qquad (6.163)$$

Сначала докажем существование кода с $k_0=1$ и $d_{FD}=\max_{FD}(m, n_0, k_0)$, для которого $G_0''^T=(1, 1, \ldots, 1)$. Пусть $k=1, \ldots, n_0-1$ — произвольное целое число. Покажем, как по коду, k-й элемент матрицы G_0'' которого равен 0, можно построить код, k-й элемент порождающей матрицы G_0'' которого равен 1 и минимальное расстояние d_{FD} которого не меньше, чем минимальное расстояние исходного кода.

Предположим, что задан некоторый код, k-й элемент порождающей матрицы G_0'' которого равен 0. Если k-е элементы всех остальных матриц G_j'', $j\neq 0$, этого кода равны 0, то k-е элементы всех блоков t_u^* в формуле (6.159) также будут равны 0. Следовательно, если k-й элемент матрицы G_0'', равный 0, заменить на 1, то минимальное расстояние d_{FD} можно увеличить.

Допустим, что k-е элементы матриц G_j'' не все равны 0. Пусть G_l'' — порождающая матрица, имеющая наименьший индекс l в множестве матриц G_j'', k-й элемент которых отличен от нуля. Если k-й элемент матрицы G_{l+j}'', $j=0, 1, \ldots, m-n$, поместить на место k-го элемента матрицы G_j'', $j=0, 1, \ldots, m-n$, то в результате k-й элемент блока t_{l+j}^* просто переместится на место k-го элемента блока t_j^*. Из определения d_{FD} и формулы (6.159) следует, что новый код имеет минимальное расстояние, по крайней мере не меньшее, чем исходный код, а k-й элемент его матрицы G_0'' равен 1. Так как k — произвольное число из множества $\{1, 2, \ldots, n_0\}$, то при $k_0=1$ существует код с минимальным расстоянием $d_{FD}=\max_{FD}(m, n_0, k_0)$ и $G_0''^T=(1, 1, \ldots, 1)$.

Следующий шаг на пути доказательства неравенства (6.163) заключается в том, чтобы показать, что для любого сверточного кода с $k_0=1$ и $G_0''^T=(1, 1, \ldots, 1)$ информационные символы $i_1, \ldots, i_m$ всегда можно выбрать так, что соответствующий им кодовый вектор $(i_0, i_1, \ldots, i_m, t_0^*, t_1^*, \ldots, t_m^*)$, $i_0=1$, будет иметь вес, не превышающий правой части (6.163). Сначала заметим, что подблок (i_0, t_0^*) вектора $(i_0, i_1, \ldots, i_m, t_0^*, t_1^*, \ldots, t_m^*)$ такой, что $(i_0, t_0^*)=(1, G_0''^T)$, т. е. является вектором веса n_0.

Далее, произвольный подблок (i_u, t_u^*), $u > 0$, представим в следующем виде:

$$(i_u, t_u^*) = \left(i_u, \sum_{j=0}^{u} i_{u-j}G_j''\right) = \left(i_u, i_uG_0'' + \sum_{j=1}^{u} i_{u-j}G_j''\right) =$$

$$= \underbrace{(i_u, i_u, \ldots, i_u)}_{n_0} + \left(0, \sum_{j=1}^{u} i_{u-j}G_j''\right). \qquad (6.164)$$

Второе слагаемое в правой части (6.164) представляет собой вектор длины n_0, не зависящий от i_u. Следовательно, векторы (i_u, t_u^*), соответствующие $i_u = 0$ и $i_u = 1$, являются дополнительными. Так как n_0 — по предположению нечетное число, то, выбрав i_u равным 0 или 1, можно добиться того, что вес n_0-мерного вектора (i_u, t_u^*) не будет превышать $(n_0 - 1)/2$. Поскольку вышеизложенное справедливо при любом $u = 1, 2, \ldots, m$, то в любом коде с $G_0''^T = (1, 1, \ldots, 1)$ всегда существует кодовое слово, вес которого не превосходит $n_0 + m(n_0 - 1)/2$. Это доказывает свойство 2.

Свойство 3. При любом четном m

$$\max_{FD}(m, n_0, k_0) \leqslant \max_{FD}(m/2, 2n_0, 2k_0). \qquad (6.165)$$

Чтобы доказать это свойство, рассмотрим код с параметрами $m = 2m'$, n_0, k_0, порождаемый матрицами G_0'', G_1'', $\ldots$, $G_{2m'}''$, и построим по нему новый код с параметрами m', $n_0' = 2n_0$, $k_0' = 2k_0$, определив порождающие матрицы последнего следующим образом:

$$G_j^* = \begin{vmatrix} G_{2j}'' & G_{2j+1}'' \\ G_{2j-1}'' & G_{2j}'' \end{vmatrix}, \qquad j = 0, 1, \ldots, m', \qquad (6.166)$$

где G_{-1}'' и G_{m+1}'' — матрицы, все элементы которых равны 0. Обозначим через

$$i_j' = (i_{2j}, i_{2j+1}) \qquad (6.167)$$

и

$$t_j' = (t_{2j}^*, t_{2j+1}^*) \qquad (6.168)$$

векторы соответственно из $k_0' = 2k_0$ информационных и $n_0' - k_0' = 2(n_0 - k_0)$ проверочных двоичных символов нового кода в момент времени j. По определению

$$t_u' = i_u'G_0^* + i_{u+1}'G_1^* + \ldots + i_{u-m}'G_m^*. \qquad (6.169)$$

Блоки t_u^* кодовых слов нового кода, как легко видеть, являются также блоками кодовых слов исходного кода, порождаемого матрицами G_l'', и могут быть найдены с помощью равенства (6.159). Согласно равенству (6.160), минимальное расстояние нового кода определяется равенством

$$d'_{FD} = \min_{\substack{i_0 \neq 0 \text{ или } i_1 \neq 0 \\ (\text{либо } i_0 \neq 0,\ i_1 \neq 0)}} W_H(i_0, i_1, \ldots, i_{m+1}, t_0^*, t_1^*, \ldots, t_{m+1}^*). \quad (6.170)$$

Можно показать, что правая часть равенства (6.170) к тому же является минимальным расстоянием d_{FD} исходного кода. Действительно, минимум W_H $(i_0, i_1, \ldots, i_{m+1}, t_0^*, t_1^*, \ldots, t_{m+1}^*)$ по всем информационным последовательностям, для которых $i_0 = 0$ (в этом случае $t_0^* = 0$) и $i_1 \neq 0$, равен минимальному расстоянию d_{FD} по определению. Так как при $i_0 \neq 0$ вектор $(i_0, \ldots, i_m, t_0^*, \ldots, t_m^*)$ не может иметь вес, меньший d_{FD}, то, очевидно, что вес вектора $(i_0, \ldots, i_m, i_{m+1}, t_0^*, \ldots, t_m^*, t_{m+1}^*)$ также не меньше d_{FD}. Следовательно, правая часть равенства (6.170) равна d_{FD}. Отсюда следует неравенство (6.162).

Свойство 4.

$$\max_{FD}(m, n_0, k_0) \leqslant \max_{FD}(m+1, n_0, k_0). \quad (6.171)$$

Это свойство очевидно.

Используя приведенные выше свойства, докажем следующую теорему.

Теорема 6.6 (Месси.) *Величина* $\max_{FD}(m, n_0, k_0)$ *удовлетворяет следующему неравенству*:

$$\max_{FD}(m, n_0, k_0) \leqslant \left[\frac{m+5}{2}\right](n_0 - k_0) + 1. \quad (6.172)$$

Доказательство. Сначала рассмотрим случай, когда $n_0 - k_0$ нечетно. Из свойств 1, 3, 4 непосредственно следует, что

$$\max_{FD}(m, n_0, k_0) \leqslant \max_{FD}(m, n_0 - k_0 + 1, 1) \leqslant$$
$$\leqslant \max\left(\left[\frac{m+1}{2}\right],\ 2n_0 - 2k_0 + 2,\ 2\right).$$

Отсюда и из свойств 1 и 2 получаем

$$\max_{FD}(m, n_0, k_0) \leqslant \max_{FD}\left(\left[\frac{m+1}{2}\right],\ 2n_0 - 2k_0 + 1,\ 1\right) \leqslant$$
$$\leqslant 2(n_0 - k_0) + 1 + \left[\frac{m+1}{2}\right](n_0 - k_0).$$

Предположим, что $(n_0 - k_0)$ четное. Тогда, согласно свойствам 1 и 2,

$$\max_{FD}(m, n_0, k_0) \leqslant \max_{FD}(m, n_0 - k_0 + 1) \leqslant$$

$$\leqslant (n_0 - k_0 + 1) + \frac{m}{2}(n_0 - k_0).$$

Подставляя $n_{FD} = (m+1)n_0$ и $R = k_0/n_0$ в (6.172), получаем следующее асимптотическое неравенство:

$$\lim_{m\to\infty} \frac{\max_{FD}(m, n_0, k_0)}{n_{FD}} \leqslant \frac{1}{2}(1 - R). \tag{6.173}$$

6.9.2. Нижняя граница Гилберта для сверточных кодов при декодировании с обратной связью

Нижняя граница Гилберта для сверточных кодов, используемых при декодировании с обратной связью, для произвольных скоростей передачи $R = k_0/n_0$ была получена Возенкрафтом [1], когда $n_0 = k_0 + 1$. В общем случае нижняя граница выводится аналогично, в частности, использованный ниже метод вывода этой границы может быть непосредственно обобщен на случай произвольных кодов.

В случае $n_0 = k_0 + 1$ блоки $\boldsymbol{t}_u^*$ представляют собой просто двоичные символы $t_u^{(k_0+1)}$, а матрицы G_j'' вырождаются в вектор-столбцы

$$\boldsymbol{g}_u''^T = (g_{1,u}^{(k_0+1)}, \ldots, g_{k_0,u}^{(k_0+1)}).$$

При этом

$$\begin{bmatrix} t_0^{(k_0+1)} \\ t_1^{(k_0+1)} \\ \cdot \\ \cdot \\ \cdot \\ t_m^{(k_0+1)} \end{bmatrix} = \begin{bmatrix} i_0, & 0, \ldots, 0 \\ i_1, & i_0, \ldots, 0 \\ \cdot & \\ \cdot & \\ \cdot & \\ i_m, & i_{m-1}, \ldots, i_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{g}_0'' \\ \boldsymbol{g}_1'' \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{g}_m'' \end{bmatrix}. \tag{6.174}$$

Вектор в левой части (6.174) назовем t-вектором, матрицу размера $(m+1) \times (m+1)k_0$ в правой части — информационной матрицей, а вектор $(i_0, i_1, \ldots, i_m)$ — информационным вектором. Вектор-строку, компонентами которого являются компоненты информационного вектора и t-вектора, связанные между собой равенством (6.174), будем называть начальным кодовым вектором. Начальный кодовый вектор состоит из $n_{FD} = (k_0+1)(m+1)$ двоичных символов, $k_0(m+1)$ из которых являются компонентами информационного вектора, а остальные $(m+1)$ — компонентами t-вектора. Говорят, что некоторый код имеет минималь-

ное расстояние d_{FD}, строго меньшее d, если вес некоторого его начального кодового вектора меньше d. Всего имеется $\sum_{j=0}^{d-1} \binom{n_{FD}}{j}$ двоичных векторов длиной n_{FD} и веса $d-1$ или менее. Далее потребуется следующая известная оценка сверху для указанной выше суммы в следующей лемме.

Лемма 6.2.

$$\sum_{j=0}^{u} \binom{n}{j} \leqslant 2^{nH(u/n)}, \tag{6.175}$$

где $H(x)$ — функция, называемая энтропией и определяемая равенством

$$H(x) = -x \log x - (1-x) \log (1-x). \tag{6.176}$$

Доказательство. Лемму 6.2 можно доказать различными способами. Здесь, следуя Парку [20], докажем ее методом математической индукции. Для этого положим

$$A(n, u) = \sum_{j=0}^{u} \binom{n}{j}, \tag{6.177}$$

$$B(n, u) = 2^{nH(u/n)} = (n/u)^u [n/(n-u)]^{n-u}. \tag{6.178}$$

1. В случае $u = 0$: $A(n, 0) = B(n, 0) = 1$ и (6.175) выполняется со знаком равенства.
2. В случае $n = 2u$: $A(2u, u) < A(2u, 2u) = 2^{2u} = B(2u, u)$.
3. В случае $0 < u < n/2$ справедливость неравенства (6.175) докажем индукцией. Предположим, что неравенство (6.175) имеет место при (n, u) и $(n, u+1)$, и докажем его справедливость при $(n+1, u+1)$. Сначала заметим, что

$$\binom{n+1}{u+1} = \binom{n}{u+1} + \binom{n}{u}$$

и, следовательно,

$$A(n+1, u+1) = A(n, u+1) + A(n, u). \tag{6.179}$$

Оценивая слагаемые в правой части (6.179) с помощью неравенства (6.175), которое по предположению индукции выполняется при данных (n, u) и $(n, u+1)$, и полагая

$$\alpha = \left[\frac{n}{n+1}\right]^n \left[\frac{u+1}{n+1}\right] \left[\frac{u+1}{u}\right]^u, \tag{6.180}$$

$$\beta = \left[\frac{n}{n+1}\right]^n \left[\frac{n-u}{n-u-1}\right]^{n-u-1} \left[\frac{n-u}{n+1}\right], \tag{6.181}$$

после несложных преобразований находим, что

$$A(n+1, u+1) \leqslant B(n+1, u+1)[\alpha+\beta]. \qquad (6.182)$$

Покажем, что $\alpha+\beta \leqslant 1$, и этим завершим доказательство леммы. Логарифмируя выражения (6.180) и (6.181) и используя неравенство $\ln x \leqslant x-1$, получаем

$$\ln \alpha \leqslant \frac{-(n-u-1)}{n+1}, \qquad (6.183)$$

$$\ln \beta \leqslant \frac{-u}{n+1}. \qquad (6.184)$$

Отсюда следует, что

$$\alpha+\beta \leqslant [e^{-(n-u-1)}+e^{-u}]^{1/n+1}. \qquad (6.185)$$

Так как $n-u-1 \geqslant 1$ и $u \geqslant 1$ при всех $1 \leqslant u \leqslant n-2$, то $e^{-(n-u-1)}+e^{-u} \leqslant e^{-1}+e^{-1} = 2/e < 1$ и, следовательно, $\alpha+\beta<1$. Лемма 6.2 доказана.

Используя неравенство (6.175), сумму $\sum_{j=0}^{d_{FD}-1} \binom{n_{FD}}{j}$ можно оценить сверху следующим образом:

$$\sum_{j=0}^{d_{FD}-1} \binom{n_{FD}}{j} \leqslant 2^{n_{FD}H(d/n_{FD})}. \qquad (6.186)$$

Так как в случае $i_0 \neq 0$ информационная матрица в (6.174) имеет ранг $m+1$, то доля кодов, таких, что заданный информационный вектор с $i_0 \neq 0$ и произвольный фиксированный t-вектор являются решением (6.174) (т. е. доля кодов, содержащих начальное кодовое слово, образованное заданным информационным вектором с $i_0 \neq 0$ и заданным t-вектором), равна $2^{(m+1)(k_0-1)}/2^{(m+1)k_0} = 1/2^{m+1}$. Следовательно, доля кодов, содержащих по крайней мере одно из начальных кодовых слов веса d_{FD} или менее, не превосходит $2^{n_{FD}H(d/n_{FD})} \times (1/2^{m+1})$ (здесь мы воспользовались неравенством (6.186)). Отсюда в свою очередь следует, что если $2^{n_{FD}H(d/n_{FD})} \times (1/2^{m+1}) < 1$, то по крайней мере один из кодов имеет минимальное расстояние d_{FD} или более. Логарифмируя последнее неравенство и учитывая, что $n_{FD} = (m+1)(k_0+1)$, $R = k_0/(k_0+1)$, указанное выше условие существования кода с минимальным расстоянием d_{FD} или более перепишем в виде $(m+1)(k_0+1)H(d_{FD}/n_{FD}) < m+1$, или, что то же самое, $H(d_{FD}/n_{FD}) < 1/(k_0+1) = 1-R$. Если этот результат обобщить на случай произвольных кодов, то получится следующая теорема.

Теорема 6.7. *Существует по крайней мере один сверточный код со скоростью передачи R, кодовым ограничением n_A и минимальным расстоянием d_{FD}, где d_{FD} — наибольшее целое число, удовлетворяющее условию*

$$H(d_{FD}/n_{FD}) > 1 - R. \tag{6.187}$$

Граница, задаваемая формулой (6.187), *называется границей Гилберта для сверточных кодов.*

6.9.3. Верхняя и нижняя границы минимального расстояния при дефинитном декодировании

Как описывалось выше, при дефинитном декодировании с целью подавления распространения ошибок решение об информационном символе, передаваемом в момент времени u, принимается по информационным символам, принятым с момента времени $u - m$ до момента $u + m$, и проверочным символам, принятым с момента времени u до момента времени $u + m$. Число этих символов называется кодовым ограничением при дефинитном декодировании n_{DD}, а именно по определению

$$n_{DD} = (2m + 1)\,k_0 + (m + 1)(n_0 - k_0). \tag{6.188}$$

Минимальное расстояние кода d_{DD} при определенном декодировании определяется как минимальное число символов в пределах кодового ограничения, в которых различаются две кодовые последовательности, соответствующие различным значениям i_0. В силу свойства линейности сверточных кодов в данном случае минимальное расстояние d_{DD} можно также выразить через веса Хэмминга кодовых последовательностей:

$$d_{DD} = \min_{i_0 \neq 0} W_H(\boldsymbol{i}_{-m}, \boldsymbol{i}_{-m-1}, \ldots, \boldsymbol{i}_m, \boldsymbol{t}_0^*, \boldsymbol{t}_1^*, \ldots, \boldsymbol{t}_m^*). \tag{6.189}$$

Если сравнить формулы (6.160) и (6.189), то сразу же можно заметить, что $d_{DD} \leqslant d_{FD}$. Следовательно, верхняя граница для d_{FD} является также верхней границей и для d_{DD}. Однако нижняя граница для d_{FD} нижней границей d_{DD} не является.

Задача нахождения нижней границы для d_{DD} была поставлена и решена Месси [18]. Метод, использованный Месси для вывода этой границы, довольно сложен и основан на использовании периодичности последовательностей на выходе регистра сдвига. Поэтому здесь приводится лишь сама граница.

Теорема 6.8 (Месси.) *Для произвольного $n_0 > k_0$ и достаточно больших n_{DD} существует по крайней мере один сверточный код*

со скоростью $R = k_0/n_0$ и кодовым ограничением n_{DD}, минимальное расстояние d_{DD} которого удовлетворяет условию

$$H\left(\frac{d_{DD}}{n_{DD}}\right) \geqslant \frac{1}{10}\,\frac{1-R}{1+R}. \tag{6.190}$$

6.10. Коды, используемые при декодировании с обратной связью

Важнейшей характеристикой сверточных кодов, исправляющих случайные ошибки, точно так же, как в блоковых кодах, является их минимальное расстояние. Однако сверточным кодам не присущи алгебраические свойства, подобные свойствам циклических кодов, и строить сверточные коды с большим минимальным расстоянием методами, подобными методам построения БХЧ-кодов, не удается. Ряд сверточных кодов с большим минимальным расстоянием для скоростей передачи $R = 1/2$ и $1/3$ был построен Бусгангом [21]. Относительно регулярный метод построения таких кодов предложен также Лином [22]. Здесь будет описан метод поиска сверточных кодов с помощью некоторых простых алгоритмов, предложенных Костелло [19], и приведены получающиеся коды. Класс построенных Костелло кодов является наиболее широким из всех известных в настоящее время. При этом Костелло ввел два важных понятия: *столбцовое расстояние*, используемое при декодировании с обратной связью, и *свободное расстояние*, используемое при последовательном декодировании.

Порождающая матрица G_∞ сверточного кода, введенная в разд. 5.2, при скорости передачи $R = 1/n_0$ имеет следующий вид:

$$\begin{bmatrix} 1, & g^{(1)}_{(2),\,0}, & \ldots, & g^{(1)}_{(n_0),\,0}, & 0, & g^{(1)}_{(2),\,1}, & \ldots, & g^{(1)}_{(n_0,\,1)}, & 0, & g^{(1)}_{(2),\,2}, & \ldots \\ \ldots, & g^{(1)}_{(n_0),\,2}, & \ldots, & 1, & g^{(1)}_{(2)\,0}, & \ldots, & g^{(1)}_{(n_0),\,0}, & 0, & g^{(1)}_{(2,\,1)}, & \ldots, & g^{(1)}_{(n_0),\,1}, \ldots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}. \tag{6.191}$$

Подматрицу матрицы G_∞, состоящую из первых $u+1$ строк и $(u+1)(n_0+1)$ столбцов, обозначим через G^u_∞:

$$G^u_\infty = \begin{bmatrix} 1, & g^{(1)}_{(2),\,0}, & \ldots, & g^{(1)}_{(n_0),\,0}, & \ldots, & 0, & g^{(1)}_{(2),\,u}, & \ldots, & g^{(1)}_{(n_0),\,u} \\ & & & & & \vdots & & & \\ & & & & & 1, & g^{(1)}_{(2),\,0}, & \ldots, & g^{(1)}_{(n_0),\,0} \end{bmatrix}. \tag{6.192}$$

Пусть

$$\boldsymbol{i}_u = [i_0,\ i_1,\ \ldots,\ i_u] \tag{6.193}$$

— информационный вектор. Введенное в разд. 6.9.1 минимальное расстояние d_{FD} сверточного кода определяется равенством

$$d_{FD} = \min_{i_0 \neq i_0'} d_H\left[\boldsymbol{i}_m G_\infty^m, \boldsymbol{i}_m' G_\infty^m\right] = \min_{i_0 \neq 0} W_H\left[\boldsymbol{i}_m G_\infty^m\right]. \quad (6.194)$$

Используемое ниже столбцовое расстояние определяется следующим образом.

Определение 6.1. *Столбцовое расстояние порядка и сверточного кода со скоростью передачи* $R = 1/n_0$ *равно*

$$d_u = \min_{i_0 \neq i_0'} W\left(\boldsymbol{i}_u G_\infty^u, \boldsymbol{i}_u' G_\infty^u\right), \qquad u = 0, 1, \ldots . \quad (6.195)$$

Сравнивая формулы (6.194) и (6.195), можно заметить, что $d_{FD} = d_m$. Введенное столбцовое расстояние обладает следующими свойствами:

1) $$d_u = \min_{i_0 \neq 0} W_H\left(\boldsymbol{i}_j G_\infty^u\right); \quad (6.196)$$

2) $$d_u \leqslant W_H(\boldsymbol{g}_u), \quad (6.197)$$

где $\boldsymbol{g}_u$ — первая строка матрицы G_∞^u;

3) $$d_u \leqslant d_{u+1}, \quad u = 0, 1, 2, \ldots . \quad (6.198)$$

Свойство 1 непосредственно следует из определений минимального расстояния кода и минимального веса. Полагая $i_0 = 1$, $i_1 = i_2 = \ldots = i_u = 0$, получаем $\boldsymbol{i}_u G_\infty^u = \boldsymbol{g}_u$; отсюда и из свойства 1 следует справедливость свойства 2. Свойство 3 следует из того, что вектор $\boldsymbol{i}_u G_\infty^u$ всегда является префиксом вектора $[\boldsymbol{i}_u, i_{u+1}]\, G_\infty^{u+1} = \boldsymbol{i}_{u+1} G_\infty^{u+1}$.

Как указывалось выше, сверточный код тем лучше, чем большее отношение d_{FD}/n_{FD} он имеет. Одним из критериев, позволяющих судить о том, является ли отношение d_{FD}/n_{FD} плохим или хорошим, является близость этого отношения к границе Гилберта, определяемой формулой (6.187). Так как число сумматоров по модулю 2, содержащихся в показанном на фиг. 6.5 Rn_A-разрядном кодере сверточного кода со скоростью $R = 1/n_0$, в точности равно $W_H(\boldsymbol{g}_m) - n_0$, то при заданных параметрах d_{FD} и n_{FD} более предпочтительными оказываются коды, которые имеют меньшее значение $W_H(\boldsymbol{g}_m)$.

а) Коды со скоростью $R = 1/2$. Коды с такой скоростью могут быть построены с помощью следующего алгоритма:

Алгоритм I.

0) Положить $g^{(1)}_{(2),\,0}=1$, $d_0=2$, $u=1$.

1) Положить $g^{(1)}_{(2),\,u}=1$.

2) Вычислить d_u. Если $d_u > d_{u-1}$, то перейти к шагу 4.

3) Положить $g^{(1)}_{(2),\,u}=0$.

4) Если $u=m$, то построение кода заканчивается; в противном случае положить $u=u+1$ и перейти к шагу 1.

Свойство I-1. $W_H(\boldsymbol{g}_u)=d_u$, $u=0, 1, \ldots, m$.

Доказательство. Из того, что $W_H(\boldsymbol{g}_0)=d_0=2$, и из свойства 2 следует, что $W_H(\boldsymbol{g}_u)\geqslant d_u$. С другой стороны, шаги 2—4 алгоритма таковы, что при $d_u > d_{u-1}$ и только в этом случае величина $W_H(\boldsymbol{g}_u)$ увеличивается на единицу, но не больше. Следовательно, $W_H(\boldsymbol{g}_u)\leqslant d_u$, а поэтому $W_H(\boldsymbol{g}_u)=d_u$.

В силу свойства 2 столбцового расстояния $W_H(\boldsymbol{g}_m)\geqslant d_m = d_{FD}$, но, как следует из свойства I-1, в данном случае величина $W_H(\boldsymbol{g}_m)$ принимает минимальное возможное значение, а это означает в свою очередь, что число сумматоров по модулю 2 в кодере также минимально.

Свойство I-2. Если $g^{(1)}_{(2),\,u}=1$, то $g^{(1)}_{(2),\,u+1}=0$ для всех $u\neq 0$.

Доказательство. Предположим, что $g^{(1)}_{(2),\,u}=1$ при некотором $u\geqslant 1$ и $g^{(1)}_{(2),\,u+1}=1$. В этом случае информационная последовательность $i_0=i_u=1$, $i_1=i_2=\ldots=i_{u-1}=i_{u+1}=0$ всегда будет порождать кодовое слово веса d_u, так что $d_{u+1}=d_u$. Однако это противоречит тому, что в силу свойства I при сделанных предположениях $d_{u+1}=W(g_{u+1})>W(g_u)=d_u$. Следовательно, если $g^{(1)}_{(2),\,u}=1$, $u\geqslant 1$, то для алгоритма I всегда $g^{(1)}_{(2),\,u+1}=0$.

Из свойства I-2 следует, что всякий раз, когда коэффициент $g^{(1)}_{(2),\,u}$ полагается равным 1, следующий коэффициент $g^{(1)}_{(2),\,u+1}$ должен равняться 0. Это свойство алгоритма I позволяет сократить число операций 1 и 2 при построении кода.

Свойство I-3. Пусть $\boldsymbol{g}_m$ — порождающий вектор, получающийся при построении кода с помощью алгоритма I. Пусть $\boldsymbol{g}'_m$ — другой порождающий вектор той же длины, что и $\boldsymbol{g}_m$, но отличный от последнего и такой, что $W_H(\boldsymbol{g}'_u)=d'_u$ при любом целом $u=0, 1, \ldots, m$, а именно $\boldsymbol{g}'_m$ — вектор, любой символ 1 которого увеличивает минимальное расстояние ровно на единицу. Тогда существует такое целое число u_0, $1\leqslant u_0\leqslant m$, что $d_i=d'_i$ при $i=0, 1, \ldots, u_0-1$ и $d_{u_0} > d'_{u_0}$.

Доказательство. Пусть u_0, $0 \leqslant u_0 \leqslant m$, — начальная точка, в которой начинают различаться порождающие векторы g_m и g'_m. Допустим, что $g^{(1)}_{(2),\,u_0} = 0$ и $g^{(1)'}_{(2),\,u_0} = 1$. Тогда $d'_{u_0} = d_{u_0} + 1 > d_{u_0}$. Это означает, что столбцовое расстояние в точке u_0 увеличивается, а следовательно, согласно алгоритму I, следовало бы положить $g^{(1)}_{(2),\,u_0} = 1$, но это противоречит предположению о том, что $g^{(1)}_{(2),\,u_0} = 0$. Следовательно, в точке u_0, где порождающие векторы $g_{m'}$ и g_m впервые начинают различаться, $g^{(1)'}_{(2),\,u_0} = 0$, $g^{(1)}_{(2),\,u_0} = 1$ и $d_{u_0} > d'_{u_0}$.

Коды, которые могут быть построены с помощью других алгоритмов, отличаются от кодов, алгоритм построения которых был описан выше, тем, что их порождающие последовательности могут не содержать символ 1 на позиции, где он мог бы быть введен.

Коды с $R = 1/2$, полученные с помощью алгоритма I, и их минимальное расстояние, удовлетворяюшее границе Гилберта, приведены в табл. 6.1.

Таблица 6.1

Коды со скоростью $R = 1/2$

u	$g^{(1)}_{(2),\,u}$	d_u	u	$g^{(1)}_{(2),\,u}$	d_u	u	$g^{(1)}_{(2),\,u}$	d_u	u	$g^{(1)}_{(2),\,u}$	d_u
0	1	2	17	0	8	34	0	12	50	0	16
1	1	3	18	0	8	35	1	13	51	0	16
2	0	3	19	0	8	36	0	13	52	0	16
3	1	4	20	1	9	37	0	13	53	1	17
4	0	4	21	0	9	38	0	13	54	0	17
5	1	5	22	0	9	39	0	13	55	0	17
6	0	5	23	0	9	40	1	14	56	1	18
7	0	5	24	1	10	41	0	14	57	0	18
8	1	6	25	0	10	42	0	14	58	0	18
9	0	6	26	0	10	43	1	15	59	0	18
10	0	6	27	1	11	44	0	15	60	0	18
11	1	7	28	0	11	45	0	15	61	0	18
12	0	7	29	0	11	46	0	15	62	1	19
13	0	7	30	0	11	47	0	15	66	0	19
14	0	7	31	1	12	48	1	16	64	0	19
15	0	7	32	0	12	49	0	16	65	1	20
16	1	8	33	0	12						

б) Коды со скоростью $R = 1/3$. Коды со скоростью $R = 1/n_0$, $n_0 > 2$, также можно получить, строя порождающие последовательности, удовлетворяющие условию $d_u = W_H(g_u)$, $u = 0, 1, \ldots, m$, путем присоединения к уже построенной последовательности g_{u-1} совокупности из $n_0 - 1$ символов, по крайней мере один из которых равен 1, если при этом не нарушается

условие $d_u = W_H(g_u)$, и совокупности из $n_0 - 1$ нулевых символов в противном случае. При этом для каждого значения u приходится определить уже $n_0 - 1$ символов $g^{(1)}_{(2),\,u}, \ldots, g^{(1)}_{(n_0),\,u}$.

В случае $n_0 = 3$ удлинить порожденную последовательность можно тремя указанными ниже способами. Для построения порождающей последовательности g_u, удовлетворяющей условию $d_u = W_H(g_u)$, при каждом значении u в первую очередь используется тот способ, который не приводит к нарушению условия $d_u = W_H(g_u)$ и в то же время позволяет повысить минимальное расстояние.

Алгоритм II.

0) Положить $g^{(1)}_{(2),\,0} = g^{(1)}_{(3),\,0} = 1$, $d_0 = 3$ и $u = 1$.

1) Положить $g^{(1)}_{(2),\,u} = 1$, $g^{(1)}_{(3),\,u} = 0$.

2) Вычислить d_u. Если $d_u > d_{u-1}$, то перейти к шагу 6.

3) Положить $g^{(1)}_{(2),\,u} = 0$, $g^{(1)}_{(3),\,u} = 1$.

4) Вычислить d_u. Если $d_u > d_{u-1}$, то перейти к шагу 6.

5) Положить $g^{(1)}_{(2),\,u} = g^{(1)}_{(3),\,u} = 0$.

6) Если $u = m$, то построение кода заканчивается; в противном случае положить $u = u + 1$ и перейти к шагу 1.

Заметим, что описанный алгоритм исключает возможность удлинения порождающей последовательности путем присоединения символов $g^{(1)}_{(2),\,u} = 1$ и $g^{(1)}_{(3),\,u} = 1$. Как показали Лин и Лайн [22], для данного алгоритма всегда $d_u \leqslant d_{u-1} + 1$.

Таблица 6.2

Коды со скоростью $R = 1/3$

u	$g^{(2)}_{(1),\,u}$	$g^{(3)}_{(1),\,u}$	d_u	u	$g^{(2)}_{(1),\,u}$	$g^{(3)}_{(1),\,u}$	d_u
0	1	1	3	14	0	0	14
1	1	0	4	15	0	0	15
2	1	0	5	16	0	0	15
3	0	1	6	17	1	0	16
4	1	0	7	18	0	1	17
5	0	1	8	19	0	0	17
6	0	1	9	20	1	0	18
7	0	1	10	21	0	0	18
8	0	0	11	22	1	0	19
9	1	0	11	23	6	1	20
10	1	0	12	24	0	0	20
11	0	0	12	25	0	0	20
12	1	0	13	26	1	0	21
13	0	0	13				

Коды, построенные с помощью алгоритма II, а также значения минимального расстояния, найденные из границы Гилберта, приведены в табл. 6.2.

в) Коды со скоростью $R = 1/4$. В этом случае для каждого значения u должны быть определены уже три символа $g^{(1)}_{(2),u}$, $g^{(1)}_{(3),u}$ и $g^{(1)}_{(4),u}$. Кроме того, следует заметить, что здесь при каждом значении u минимальное расстояние может увеличиться не только на 1, но и на 2. Как и все предыдущие алгоритмы, описанный ниже алгоритм позволяет построить класс порождающих последовательностей g_u, удовлетворяющих условию $d_u = W_H(g_u)$, путем присоединения к построенной последовательности g_{u-1} совокупности трех символов $g^{(1)}_{(2),u}$, $g^{(1)}_{(3),u}$, $g^{(1)}_{(4),u}$, один из которых равен 1, если при этом нарушается условие $d_u = W_H(g_u)$, и трех символов 0 в противном случае.

Алгоритм III.

0) Положить $g^{(1)}_{(2),0} = g^{(1)}_{(3),0} = g^{(1)}_{(4),0} = 1$, $d_0 = 4$, $j = 1$.

1) Положить $g^{(1)}_{(2),u} = g^{(1)}_{(3),u} = 1$, $g^{(1)}_{(4),u} = 0$, $i = 1$ и перейти к шагу 8.

2) Положить $g^{(1)}_{(3),u} = 0$, $g^{(1)}_{(4),u} = 1$, $i = 2$ и перейти к шагу 8.

3) Положить $g^{(1)}_{(2),u} = 0$, $g^{(1)}_{(3),u} = 1$, $i = 3$ и перейти к шагу 8.

4) Положить $g^{(1)}_{(3),u} = 0$, $i = 4$ и перейти к шагу 8.

5) Положить $g^{(1)}_{(3),u} = 1$, $g^{(1)}_{(4),u} = 0$, $i = 5$ и перейти к шагу 8.

6) Положить $g^{(1)}_{(2),u} = 1$, $g^{(1)}_{(3),u} = 1$, $i = 6$ и перейти к шагу 8.

7) Положить $g^{(1)}_{(2),u} = 0$ и перейти к шагу 9.

8) Вычислить d_u. Если $d_u = d_{u-1}$, то перейти к $i + 1$.

9) Если $u = m$, то построение кода заканчивается; в противном случае положить $u = u + 1$ и перейти к шагу 1.

Коды, построенные с помощью этого алгоритма, приведены в табл. 6.3.

6.11. Коды, используемые при последовательном декодировании

При последовательном и других аналогичных методах декодирования декодер принимает решение о том или ином переданном символе, основываясь не обязательно лишь на символах, лежащих в пределах кодового ограничения. Как мы видели выше в разд. 6.2, в процессе поиска правильного пути на

Таблица 6.3

Коды со скоростью $R = 1/4$

u	$g^{(2)}_{(1),u}$	$g^{(3)}_{(1),u}$	$g^{(4)}_{(1),u}$	d_u	u	$g^{(2)}_{(1),u}$	$g^{(3)}_{(1),u}$	$g^{(4)}_{(1),u}$	d_u
0	1	1	1	4	14	0	1	0	21
1	1	1	0	6	15	0	0	1	22
2	1	0	1	8	16	0	0	1	23
3	0	0	1	9	17	0	0	1	24
4	0	1	0	10	18	0	1	0	25
5	0	0	1	11	19	0	1	0	26
6	0	0	1	12	20	1	0	0	27
7	1	0	1	14	21	0	0	1	28
8	0	0	1	15	22	0	0	1	29
9	0	1	0	16	23	0	0	1	30
10	0	1	0	17	24	0	1	0	31
11	0	0	0	17	25	0	0	1	32
12	1	1	0	19	26	0	1	0	33
13	0	0	1	20	27	0	1	0	34

кодовом дереве декодер может обследовать пути, длина которых намного превышает кодовое ограничение. В подобных случаях параметр d_{FD}, характеризующий расстояние Хэмминга между кодовыми последовательностями в пределах одного кодового ограничения, не очень интересен. Другой более подходящей в подобных случаях характеристикой кода является так называемое свободное расстояние, введенное Месси.

Определение 6.2. *Свободное расстояние* $d_{\text{своб}}$ *равно* d_∞, *т. е. столбцовому расстоянию порядка* $u = \infty$. Свободное расстояние обладает следующими свойствами:

Свойство 1. Для любого u $d_u \leqslant d_{\text{своб}} \leqslant W_H(\boldsymbol{g})$ и, в частности, $d_m = d_{\text{мин}} \leqslant d_{\text{своб}} \leqslant W_H(\boldsymbol{g})$, где $\boldsymbol{g}$ — порождающий вектор кода.

Доказательство. Неравенство $d_u \leqslant d_\infty = d_{\text{своб}}$ следует непосредственно из свойств столбцового расстояния. Неравенство $d_{\text{своб}} = d_\infty \leqslant W_H(\boldsymbol{g})$ является следствием свойства 2 столбцового расстояния.

Свойство 2. Для всех кодов, которые строятся с помощью алгоритмов I, II и III, $d_{\text{своб}} = d_{FD} = d_m$.

Доказательство. Справедливость свойства 2 непосредственно следует из того, что для всех кодов, которые строятся с помощью указанных алгоритмов, $d_{FD} = d_m = W_H(\boldsymbol{g}_m) = W_H(\boldsymbol{g})$.

Свойство 3. Для систематических кодов со скоростью передачи $R = 1/n_0$ свободное расстояние $d_{\text{своб}}$ равно $d_{(n_0-1)(m+1)\,m}$.

Доказательство. Минимальное расстояние кода не может быть достигнуто на кодовых последовательностях, информационные подпоследовательности которых имеют ненулевой первый символ ($i_0 \neq 0$), но содержат m или более следующих друг за другом символов 0. Это связано с тем, что при наличии m и более следующих друг за другом символов 0 в информационной подпоследовательности вес соответствующей кодовой последовательности можно увеличить на 1. Согласно свойству 1, свободное расстояние $d_{\text{своб}}$ систематического кода со скоростью $R = 1/n_0$ не может превышать число единиц в порождающей последовательности, которое в свою очередь не превышает $(n_0 - 1)(m + 1) + 1$. Кодовая последовательность, на которой достигается минимальное расстояние, должна содержать по крайней мере один символ 1 в 0-м подблоке и по крайней мере один символ 1 в любых следующих друг за другом подблоках. Это означает, что все такие кодовые слова с $i_0 \neq 0$ в пределах первых $(n_0 - 1)(m + 1)m$ подблоков имеют вес не менее $(n_0 - 1)(m + 1) + 1$. Следовательно, $d_{\text{своб}} = d_{(n_0-1)(m+1)\,m}$.

Свободное расстояние является характеристикой кода, которая представляет интерес не только в тех случаях, когда рассматривается гипотетический декодер с $u = \infty$, но и при последовательном декодировании, когда перед принятием решения о том или ином переданном символе могут обследоваться последовательности длиной в несколько кодовых ограничений.

Коды с большими значениями $d_{\text{своб}}$ могут быть построены с помощью описанного ниже алгоритма, представляющего собой модификацию алгоритма I.

Алгоритм IV (предполагается, что $L \geqslant m$).

0) Положить $g^{(1)}_{(2),\,0} = 1$, $D_0 = 2$, $u = 1$.

1) Положить $g^{(1)}_{(2),\,u} = 1$.

2) Вычислить d_L. Если $d_L > D_{u-1}$, то положить $D_u = d_L$ и перейти к шагу 4.

3) Положить $g^{(1)}_{(2),\,u} = 0$, $D_u = D_{u-1}$.

4) Если $u = m$, то построение кода заканчивается; в противном случае положить $u = u + 1$ и перейти к шагу 1.

Коды, построенные с помощью этого алгоритма, обладают следующими свойствами:

Свойство IV-1. $W_H(g_u) = D_u$ для всех u.

Свойство IV-2. Если на шаге 2 оказывается, что $d_L > D_{u-1}$, то $D_u = D_{u-1} + 1$.

Теорема 6.9. *При всех* u: $W_H(g_u) = D_u < d_{\text{своб}}$, *где* $d_{\text{своб}}$— *свободное расстояние «усеченного» кода с памятью на* m *моментов времени.*

Доказательство. Соотношение $W_H(g_u) = D_u \leqslant W_{\text{своб}}$ следует из свойства IV-1 и свойства 1 свободного расстояния. Из свойства 1 свободного расстояния также следует, что $d_{\text{своб}} < W_H(g_u)$ и, следовательно, $d_{\text{своб}} = d_H(g_u)$ при всех u.

Таблица 6.4

Коды, построенные с помощью алгоритма IV

u	$g^{(2)}_{(1),\,u}$	d	u	$g^{(2)}_{(1),\,u}$	d	u	$g^{(2)}_{(1),\,u}$	d
0	1	2	7	1	7	14	0	10
1	1	3	8	0	7	15	0	10
2	1	4	9	1	8	16	0	10
3	0	4	10	0	8	17	0	10
4	1	5	11	0	8	18	1	11
5	1	6	12	1	9	19	1	12
6	0	6	13	1	10	20	1	13

Коды, построенные с помощью алгоритма IV, приведены в табл. 6.4. Костелло вычислил для сравнения вероятности ошибок при последовательном декодировании в случае $R = R_{\text{выч}}$ для двух кодов: кода с $d_{FD} = d_{\text{своб}} = 13$ и кода с $d_{FD} = 13$ и $d_{\text{своб}} = 17$. Оказалось, что в первом случае $p_e = 0{,}073$, а во втором случае $p_e = 0{,}017$. Это показывает, что при последовательном декодировании коды с большими значениями $d_{\text{своб}}$ являются более предпочтительными.

Задачи и упражнения

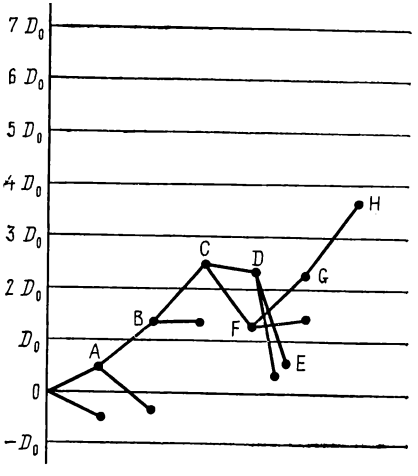
1. На фиг. 6.19 приведен пример поиска правильного пути на кодовом дереве с помощью алгоритма Фано. Для этого примера заполните приведенную ниже таблицу.

Узел	Текущее значение порога	Действие декодера	Узел	Текущее значение порога	Действие декодера
O			*C*		
A			*D*		
B			*C*		
C			*F*		
D			*G*		
C			*H*		

2. Для двоичного симметричного канала с $p = 0{,}02$ найти верхнюю границу среднего числа операций в случае, если $U = 0{,}95R_{\text{выч}}$, $R = 1/3$ и $n_0 = 2$ (постоянным членом пренебречь).

3. Для двоичного симметричного канала с $p = 0{,}02$, сверточного кода со скоростью $R = 1/3$ и кодовым ограничением $n_A = 200$ при $U = 0{,}95R_{\text{выч}}$.

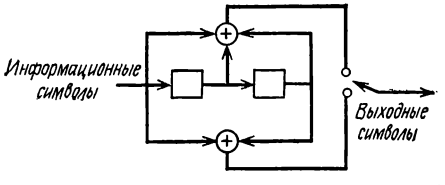
а) вычислить значение верхней границы для вероятности ошибки при использовании алгоритма Фано;



Фиг. 6.19. Пример поиска пути с помощью алгоритма Фано.

б) вычислить вероятность переполнения буфера при $\sigma = 30$ и $B = 2000$;

в) рассмотреть передачу со скоростью 4800 бит/с и вычислить приближенные значения частоты возникновения переполнения буфера и частоты ошибок.



Фиг. 6.20. Сверточный кодер кода с $n_0 = 2$, $k_0 = 1$ и $m = 2$.

4. Для кодов с относительно небольшим кодовым ограничением, используя диаграмму переходов из состояния в состояние, можно сравнительно просто провести анализ алгоритма Витерби [27].

а) Построить кодовое дерево кода, порождаемого сверточным кодером, показанным на фиг. 6.20 (до глубины, когда уже обнаруживается периодическая структура кода).

б) Используя четыре состояния упомянутой выше периодической структуры, построить диаграмму переходов из состояния в состояние.

в) Показать, что при декодировании описанного выше кода с помощью алгоритма Витерби вероятность p_k возникновения ошибки при сравнении правильного пути с любым другим путем, находящимся от него на расстоянии k, определяется равенством

$$p_k = \begin{cases} \sum\limits_{e=(k+1)/2}^{k} \binom{k}{e} p^e (1-p)^{k-e}, & k \text{ — четное число,} \\ \frac{1}{2}\binom{k}{k/2} p^{k/2}(1-p)^{k/2} + \sum\limits_{e=k/2+1}^{k} \binom{k}{e} p^e (1-p)^{k-e}, & \\ & k \text{ — нечетное число.} \end{cases}$$

г) Доказать справедливость следующего неравенства: $p_k < 2^k p^{k/2}$, где p — вероятность перехода символа 0 в символ 1 (или 1 в 0) в двоичном симметричном канале.

д) Показать, что вероятность ошибки на символ для рассматриваемого кода можно оценить сверху следующим образом:

$$p_B < \sum_{k=5}^{\infty} (k-4)\, 2^{k-5} [4p(1-p)]^{k/2}.$$

5. Доказать неравенства (6.53)—(6.56).

6. Пусть $q_1, q_2, \ldots, q_J$ — распределение вероятностей на выходе дискретного канала без памяти с K входными и J выходными символами. Если переходные вероятности этого канала таковы, что

$$p(j \mid k) = q_j (1 + \varepsilon_{jk}),$$

и $|\varepsilon_{jk}| \ll 1$ для всех j и k, то этот канал называется *каналом с сильным шумом.*

а) Показать, что для блоковых кодов функция надежности

$$E(R) = \max_{0<\nu\leqslant 1} [-\nu R + \max_{Q} E_0(\nu, Q)]$$

для канала с сильным шумом имеет следующий вид [3]:

$$E(R) \approx (\sqrt{C} - \sqrt{R})^2, \quad R \geqslant \frac{C}{4},$$

$$E(R) \approx \frac{C}{2} - R, \quad R < \frac{C}{4},$$

где C — пропускная способность канала.

б) Показать, что для сверточных кодов функция надежности канала с сильным шумом определяется равенствами [4]

$$E(R) \approx C/2, \qquad 0 \leqslant R \leqslant C/2,$$
$$E(R) \approx C - R, \quad C/2 \leqslant R < C.$$

7. Месси и др. [28], исследуя коды со скоростью $R = 1/n_0$, получили ряд интересных результатов, которые сформулированы ниже в виде задач.

а) Пусть

$$I(D) = i_0^{(1)} + i_1^{(1)} D + i_2^{(1)} D^2 + \ldots$$

— информационная последовательность. Показать, что если порождающий многочлен кода имеет вид

$$G(D) = G_{(1)}^{(1)}(D^{n_0}) + G_{(2)}^{(1)}(D^{n_0}) + \ldots + D^{n_0-1} G_{(n_0)}^{(1)}(D^{n_0}),$$

то кодовая последовательность $T(D)$ будет определяться равенством

$$T(D) = I(D^{n_0})\, G(D).$$

б) Пусть $g(x)$ — порождающий многочлен циклического кода нечетной длины над полем $GF(2^r)$. Показать, что свободное расстояние $d_{\text{своб}}$ 2^r-ичного сверточного кода со скоростью $R = 1/n_0$ (где $n_0 = 2m$ и m — произвольное целое положительное число), порождаемого многочленом $G(D) = g(D)$, удовлетворяет неравенству

$$d_{\text{своб}} \geqslant \min\{d_g,\ 2d_h\},$$

где d_g — минимальное расстояние циклического кода, порождаемого многочленом $g(x)$, а $d(h)$ — минимальное расстояние циклического кода, порождаемого многочленом $h(x) = (x^n + 1)/g(x)$.

в) Показать, что свободное расстояние $d_{\text{своб}}$ 2^r-ичного сверточного кода со скоростью $R = 1/n_0$, $n_0 = 4m$, порождаемого многочленом $G(D) = g(D)^2 + h(D)^2$, удовлетворяет неравенству

$$d_{\text{своб}} \geqslant \min\{d_g + d_h,\ 3d_g,\ 3d_h\}.$$

7. Реализация и применение кодов, исправляющих ошибки

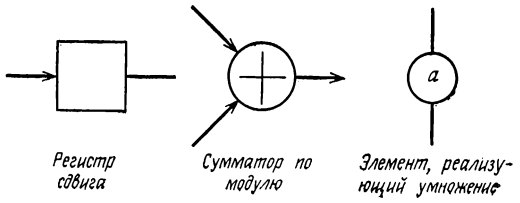
7.1. Реализация кодов, исправляющих ошибки

Кодирующие и декодирующие устройства могут быть построены из показанных на фиг. 7.1 регистров сдвига, сумматоров по модулю 2 и элементов, реализующих умножение. В результате прогресса интегральной техники, в частности техники изготовления больших интегральных схем, стоимость этих дискретных элементов в настоящее время быстро снижается.

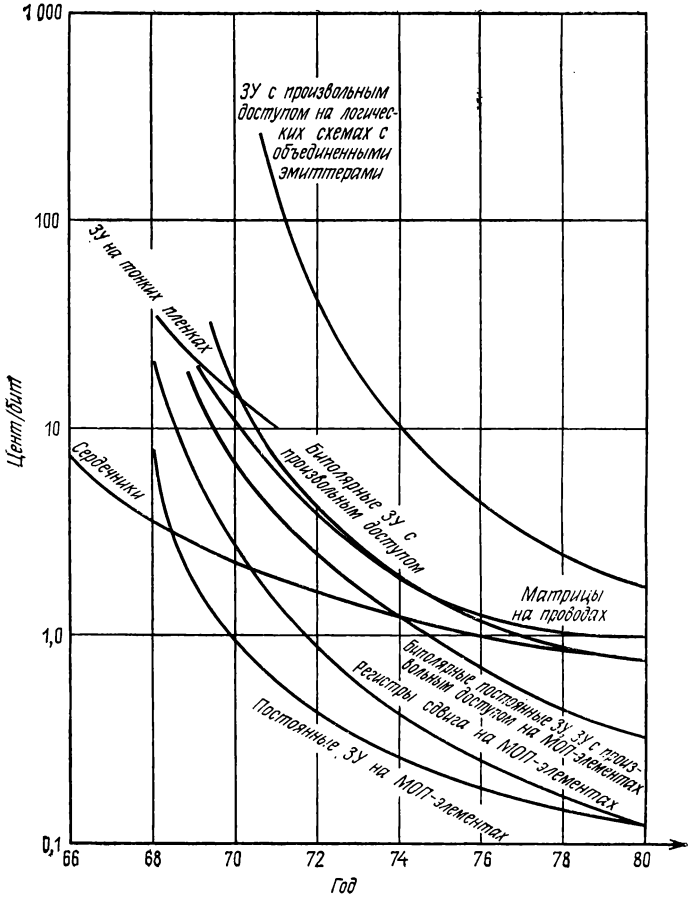
На фиг. 7.2 показано, например, каким образом происходило и происходит снижение стоимости полупроводниковой памяти. Если стоимость полупроводникового триггера вначале составляла около 10 долл., то в настоящее время стоимость памяти составляет несколько центов за 1 бит. Ожидается, что к концу семидесятых годов она достигнет 0,1 цента за 1 бит. Поэтому, если на начальном этапе развития теории кодирования реализация кодирования и декодирования с помощью аппаратурных средств из-за их сложности и высокой стоимости представлялась почти нереальной, то сейчас можно привести большое число примеров, когда аппаратура для кодирования и декодирования используется в реальных системах (главным образом в системах связи). Как правило, эта аппаратура строится из обычных широко используемых интегральных и так называемых больших интегральных схем и выполняет относительно простые стандартные операции.

Кодирование и декодирование могут быть реализованы как с помощью специализированных устройств, так и в некоторой области скоростей передачи программными методами на универсальных ЭВМ. В последние годы для программной реализации стали широко использоваться мини-ЭВМ. Для выполнения простых стандартных операций относительно просто могут быть построены небольшие специализированные вычислительные устройства. При передаче данных между вычислительными машинами для выполнения кодирования и декодирования можно использовать также часть вычислительной мощности высокоскоростных ЭВМ с большой емкостью памяти.

Применение программных методов позволяет снизить вес кодеров и декодеров. Например, если строить специализированное устройство для декодирования БЧХ-кода с большим числом исправляемых ошибок t, то при использовании алгоритма Питерсона и алгоритма Берлекэмпа — Месси приходится приме-



Фиг. 7.1. Элементы, используемые для построения кодеров и декодеров.



Фиг. 7.2. Снижение стоимости полупроводниковых запоминающих устройств.

нять сложные устройства. В подобных случаях программные методы являются эффективным средством снижения сложности устройств.

В этом разделе будут рассматриваться главным образом стандартные схемы, используемые при аппаратурной реализации кодирования и декодирования, а также ряд других вопросов, связанных с реализацией БЧХ-кодов и кодов, допускающих пороговое декодирование.

7.1.1. Стандартные схемы

Сначала рассмотрим простой пример, а именно показанную на фиг. 7.3 схему умножения фиксированного многочлена

$$b(X) = 1 + X^2 + X^3 + X^5 + X^7 \tag{7.1}$$

на входной многочлен

$$a(X) = a_0 + a_1X + \ \ldots\ + a_mX^m. \tag{7.2}$$

Все ячейки памяти показанного на фиг. 7.3 регистра сдвига вначале содержат нулевые символы. Будем считать, что на вход регистра сдвига первыми подаются старшие члены входного многочлена (т. е. члены с высшими степенями неизвестного). Член a_mX^m, поступивший на вход регистра сдвига, без изменений передается на выход. Точно так же без изменений передается на выход схемы и $a_{m-1}X^{m-1}$. После того как на вход регистра сдвига поступит $a_{m-2}X^{m-2}$ с помощью крайнего левого сумматора, он складывается с поступившим ранее a_mX^m, и их сумма

$$a_{m-2}X^{m-2}X^7 + a_mX^mX^5 = (a_{m-2} + a_m)\,X^{m+5} \tag{7.3}$$

поступает на выход схемы. Точно так же вычисляются и последующие члены искомого произведения

$$\begin{aligned}(1 + X^2 + X^3 + X^5 + X^7)(a_0 + a_1X + \ \ldots\ + a_mX^m) = \\ = a_0 + a_1X + (a_2 + a_0)\,X^2 + \ \ldots\ + (a_{m-2} + a_m)\,X^{m+5} + \\ + a_{m-1}X^{m+6} + a_mX^{m+7}.\end{aligned} \tag{7.4}$$

В общем случае умножение входного многочлена

$$a(X) = a_0 + a_1X + \ \ldots\ + a_mX^m$$

на фиксированный многочлен

$$b(X) = b_0 + b_1X + \ \ldots\ + b_nX^n$$

можно выполнить аналогично. Соответствующая схема показана на фиг. 7.4. Существует еще один тип схем, предназначенных для нахождения указанного выше произведения. Вначале рассмотрим пример такой схемы, выполняющей умножение вход-

ного многочлена $a(X)$ на фиксированный многочлен

$$b(X) = 1 + X^2 + X^3 + X^5 + X^7.$$

Схема умножения в данном случае имеет вид, показанный на фиг. 7.5. При поступлении $a_m X^m$ на вход регистра сдвига это слагаемое без изменений передается на выход схемы как $a_m X^{m+7}$. Одновременно $a_m X^m$ вводится во 2, 4, 5 и 7-й разряды регистра сдвига. Точно так же со входа без изменений на выход схемы передается $a_{m-1}X^{m-1}$ (как $a_{m-1}X^{m+6}$); при этом $a_{m-1}X^{m-1}$ вводится в регистр сдвига точно так же, как $a_m X^m$. Далее после того, как на вход поступит $a_{m-2}X^{m-2}$, член $a_m X^m$, поступивший на два момента времени ранее во 2-ю справа ячейку регистра сдвига, складывается с помощью крайнего правого сумматора с $a_{m-2}X^{m-2}$ и полученная сумма поступает на выход схемы как $(a_m + a_{m-2})X^{m+5}$. Точно так же схема работает при поступлении последующих членов входного многочлена. В общем случае схема, выполняющая умножение указанным выше образом, имеет вид, показанный на фиг. 7.6.

Так как схемы умножения описанных выше двух типов содержат одинаковое число разрядов регистров сдвига и одинаковое число сумматоров по модулю 2, то на первый взгляд с точки зрения сложности они представляются эквивалентными. Однако в схеме, изображенной на фиг. 7.6, сумматоры по модулю 2 располагаются между разрядами регистра сдвига, тогда как в схеме, изображенной на фиг. 7.4, сумматоры располагаются отдельно от регистра сдвига. При использовании интегральных схем для построения таких устройств умножения многочленов схема, изображенная на фиг. 7.4, является более предпочтительной, поскольку она может быть проще реализована.

Рассмотрим далее схему, выполняющую деление многочленов. Сначала рассмотрим пример такой схемы, а именно показанную на фиг. 7.7 схему, предназначенную для деления входного многочлена $a(X)$ на многочлен

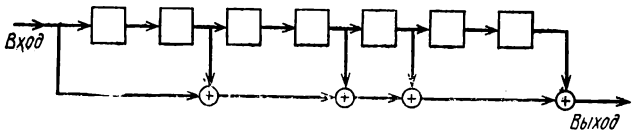
$$b(X) = 1 + X^2 + X^3 + X^5 + X^7.$$

Пусть

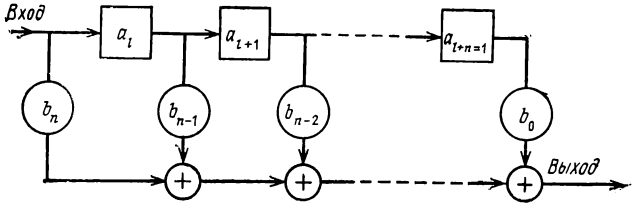
$$a(X) = 1 + X + X^3 + X^4 + X^7 + X^8 + X^{10}.$$

Деление $a(X)$ на $b(X)$ «углом» проводится следующим образом:

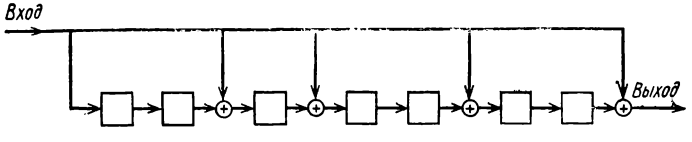
$$\begin{array}{l|l}
X^{10}+0+X^8+X^7+0+0+X^4+X^3+0+X+1 & |X^7+0+X^5+0+X^3+X^2+0+1 \\
\underline{X^{10}+0+X^8+0+X^3+X^5+0+X^3} & |X^3+0+0+1 \\
0+0+0+X^7+X^6+X^5+X^4+0+0+X+1 & \\
\underline{X^7+0+X^5+0+X^3+X^2+0+1} & \\
0+X^6+0+X^4+X^3+X^2+X+0 &
\end{array}$$



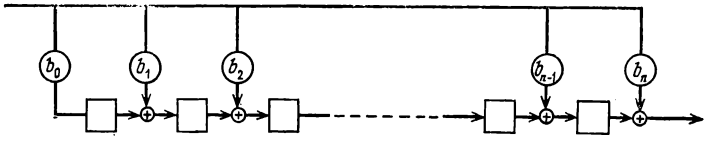
Фиг. 7.3. Схема умножения на многочлен $b(X) = 1 + X^2 + X^3 + X^5 + X^7$.



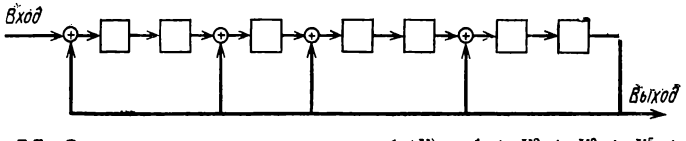
Фиг. 7.4. Схема умножения в общем случае.



Фиг. 7.5. Схема умножения на многочлен $b(X) = 1 + X^2 + X^3 + X^5 + X^7$.

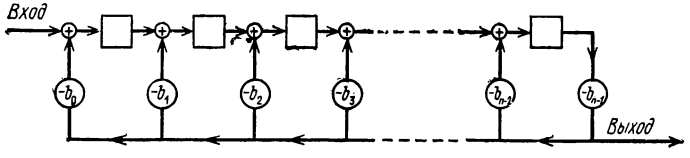


Фиг. 7.6. Схема умножения в общем случае.



Фиг. 7.7. Схема деления на многочлен $b(X) = 1 + X^2 + X^3 + X^5 + X^7$.

При делении $a(X)$ на $b(X)$ с помощью схемы, показанной на фиг. 7.7, сначала в регистр сдвига последовательно вводятся коэффициенты входного многочлена при $X^{10}, \ldots, X^4$. При поступлении на вход регистра коэффициента при X^3 коэффициент при X^{10} на входе поступает на выход схемы как коэффициент при X^3 частного. Одновременно коэффициент при X^{10} с выхода регистра подается на входы трех сумматоров по модулю 2, установленных на выходах 3, 4 и 5-й слева ячеек памяти регистра сдвига, и складывает их с выходными символами этих ячеек. При следующем сдвиге коэффициент 0 при X^9 входного многочлена выводится как 0 при X^2 частного. При следующем сдвиге символ 0, получающийся в результате сложения коэффициента 1 при X^8 входного многочлена с коэффициентом 1 при



Фиг. 7.8. Схема деления в общем случае.

X^{10}, поступающим по цепи обратной связи, выводится как коэффициент 0 при X^1 частного. После ввода всех символов входного многочлена в регистре остается остаток $X + X^2 + X^3 + X^4 + 0 + X^6$ (табл. 7.1).

В общем виде схема, выполняющая деление входного многочлена на некоторый фиксированный многочлен, изображена на фиг. 7.8.

7.1.2. Кодеры циклических кодов

Все кодовые слова циклического кода делятся на порождающий многочлен кода $g(X)$. Если циклический код является к тому же систематическим, то в каждом кодовом слове можно выделить информационные символы и следующие за ними избыточные символы. Поэтому при использовании циклических кодов сначала по каналу связи передаются информационные символы и одновременно эти символы вводятся в схему деления многочленов, описанную в предыдущем разделе. После того как все информационные символы будут введены в регистр сдвига, в последнем оказывается остаток от деления информационного многочлена на порождающий многочлен. Если этот остаток передать по каналу связи вслед за информационными символами, то полная переданная последовательность будет делиться на

$g(X)$, т. е. будет кодовым словом. Схема кодера циклического кода показана на фиг. 7.9.

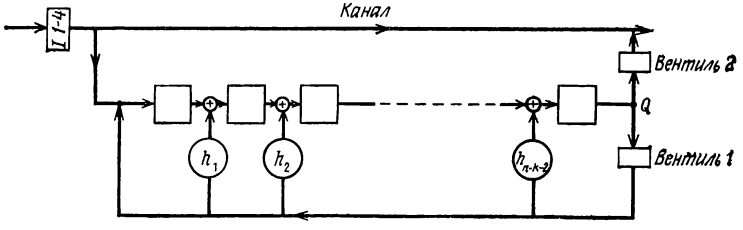
Пусть

$$f(X) = f_0 + f_1 X + f_2 X^2 + \ldots + f_{n-1} X^{n-1} \tag{7.5}$$

— кодовое слово циклического кода. Так как все кодовые слова циклического кода делятся на порождающий многочлен $g(X)$, то $f(X)$ можно представить в виде

$$f(X) = g(X)\, q(X).$$

Если код является систематическим, то информационными символами будут f_{n-k}, f_{n-k+1}, ..., f_{n-1}, а $n-k$ проверочными символами будут f_0, f_1, ..., f_{n-k-1}. Если умножить $f(X)$ на прове-



Фиг. 7.9. Кодер циклического кода.

рочный многочлен кода $h(X)$ и воспользоваться равенством $g(X)h(X) = X^n + 1$, то получим

$$f(X)h(X) = g(X)q(X)h(X) = X^n q(X) + q(X). \tag{7.6}$$

Так как $q(X)$ — многочлен степени $k-1$ или менее, то члены $X^k, X^{k+1}, \ldots, X^{n-1}$ в правую часть (7.6) не входят. Следовательно, коэффициенты при X^k, X^{k+1}, ..., X^{n-1} в произведении $f(X)h(X)$ можно приравнять нулю:

$$\sum_{i=0}^{k} h_i f_{n-i-j} = 0, \quad 1 \leqslant j \leqslant n-k \tag{7.7}$$

(здесь $h_0 = h_k = 1$). Отсюда следует, что

$$f_{n-k-j} = \sum_{i=0}^{k-1} h_i f_{n-i-j}, \quad 1 \leqslant j \leqslant n-k. \tag{7.8}$$

Последние равенства дают способ вычисления проверочных символов f_{n-k-1}, f_{n-k-2}, ..., f_0 многочлена $f(X)$. Схема кодирования, реализующая этот метод вычисления проверочных символов, показана на фиг. 7.9.

Таблица 7.1

Состояния регистра сдвига

Число сдвигов l	Состояние после l сдвигов	Символ на выходе после l сдвигов	Символы обратной связи после l сдвигов	Символ на выходе при l-м сдвиге
0	0000000	0	—	—
1	1000000	0	0000000	1
2	0100000	0	0000000	0
3	1010000	0	0000000	1
4	1101000	0	0000000	1
5	0110100	0	0000000	0
6	0011010	0	0000000	0
7	1001101	1 (частное)	0000000	1
8	0111100	0 (частное)	1011010	1
9	0011110	0 (частное)	0000000	0
10	1001111	1 (частное)	0000000	1
11	| 0111101 |	1	1011010	1
	Остаток			

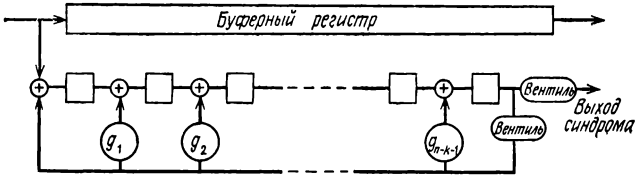
Сначала вентиль 1 закрыт, а вентиль 2 открыт. Информационные символы передаются по каналу и одновременно вводятся в регистр сдвига. После того как все k информационных символов будут введены в регистр, вентиль 1 открывается, а вентиль 2 закрывается. В этот момент в точке Q, показанной на фиг. 7.9, появляется первый проверочный символ f_{n-k-1}. При очередном сдвиге содержимого регистра первый проверочный символ вводится в крайний левый разряд регистра и одновременно передается по каналу. При этом в точке Q появляется второй проверочный символ. Точно так же формируются, вводятся в регистр сдвига и передаются по каналу все $n - k$ проверочных символов. После этого вентиль 1 закрывается, вентиль 2 открывается и в регистр сдвига вводятся следующие k информационных символов.

В тех случаях, когда допустимо использование несистематических кодов для реализации кодирующего устройства, можно воспользоваться описанной в предыдущем разделе схемой умножения многочленов. Заметим что с точки зрения простоты реализации более предпочтительными часто оказываются схемы умножения многочленов, изображенные на фиг. 7.4 и использующие сумматоры.

7.1.3. Декодеры циклических кодов

Если кодеры обычно относительно просты и однотипны, то декодеры, как правило, гораздо сложнее кодеров и их структура сильно зависит от используемых кодов. В этом разделе сначала рассматриваются простейшие декодеры, а именно декодеры кодов, обнаруживающих ошибки. Далее рассматриваются декодеры БХЧ-кодов, декодеры для кодов, допускающих пороговое декодирование, декодеры для кодов, исправляющих пачки ошибок, и некоторые другие декодеры.

а) Декодеры, обнаруживающие ошибки. Декодер этого типа состоит обычно из буферного регистра, предназначенного для хранения принятого кодового слова, и схемы деления принятого слова на порождающий многочлен кода $g(X)$. Если полученный в результате деления остаток равен нулю, то считается,



Фиг. 7.10. Декодер, обнаруживающий ошибки.

что ошибок при передаче не было. В противном случае, т. е. если хотя бы один из коэффициентов остатка отличен от нуля, принимается решение о том, что произошла ошибка при передаче; при этом прием прекращается и на передающую сторону посылается запрос на повторную передачу этого кодового слова. В показанном на фиг. 7.10 декодере принятое кодовое слово одновременно вводится в буферный регистр и изображенную под регистром схему деления многочленов. После введения принятого слова в схему деления в регистре сдвига последней оказывается остаток от деления принятого слова на порождающий многочлен кода; в зависимости от того, равен ли остаток нулю или нет, принимается решение об отсутствии или наличии ошибок в принятом слове.

Эта схема обнаружения ошибок очень проста, и устройства обнаружения ошибок с этой схемой широко используются в реальных системах. Следует заметить, что при использовании этой схемы необходим канал обратной связи, буферное запоминающее устройство для хранения данных и ряд других устройств.

б) Декодер БЧХ-кода. Как уже указывалось выше, БЧХ-коды могут иметь достаточно большое кодовое расстояние. Одна-

ко их техническая реализация оказывается сравнительно сложной.

Алгоритм декодирования БЧХ-кода включает следующие пять операций:

I. Вычисление синдрома.

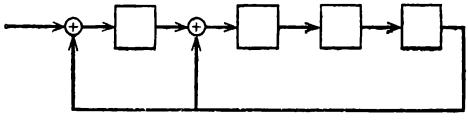
II. Вычисление многочлена, имеющего своими корнями локаторы ошибок.

III. Нахождение корней этого многочлена.

IV. Вычисление значений ошибок (в двоичном случае эта операция не нужна).

V. Исправление ошибок.

В двоичном случае операции III и V могут выполняться одновременно с помощью описанного ниже алгоритма Ченя.



Фиг. 7.11. Схема для вычисления $r(\alpha)$.

При декодировании БЧХ-кодов сначала находится синдром; эта операция может быть реализована с помощью схемы, осуществляющей вычисления в поле Галуа. Чтобы найти синдром, необходимо для принятого слова

$$r(X) = r_0 + r_1 X + \ \ldots \ + r_{n-1} X^{n-1} \tag{7.9}$$

вычислить его значение $r(\alpha^j)$ в некоторых точках α^j. Эту операцию можно выполнить посредством следующей последовательной схемы.

Рассмотрим схему вычисления синдрома для кода, корни порождающего многочлена которого лежат в поле $GF(2^4)$. Сначала рассмотрим схему вычисления $r(\alpha)$. Пусть α — примитивный элемент поля $GF(2^4)$, являющийся корнем многочлена $X^4 + X + 1$. Если в крайний левый разряд регистра схемы, изображенной на фиг. 7.11, ввести символ 1 и начать его сдвигать, то, как легко проверить, благодаря цепи обратной связи в разрядах регистра последовательно будут формироваться двоичные представления $\alpha^2, \alpha^3, \ldots$. Таким образом, чтобы вычислить

$$r(\alpha) = r_0 + r_1 \alpha + \ \ldots \ + r_{n-1} \alpha^{n-1},$$

в рассмотренную схему следует последовательно ввести символы $r_{n-1}, r_{n-2}, \ldots$.

Далее рассмотрим случай, когда $j \neq 1$. На фиг. 7.12 приведен пример схемы вычисления $r(\alpha^3)$ с помощью регистра сдвига с четырьмя разрядами [α — примитивный элемент поля

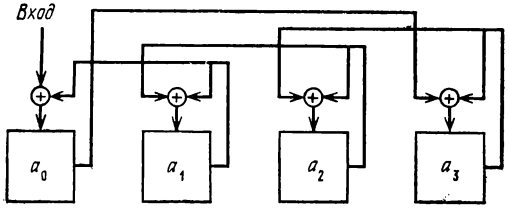
$GF(2^4)$]. Так как, согласно таблице умножения в поле $GF(2^4)$,

$$1 \cdot \alpha = \alpha^3,$$
$$\alpha \cdot \alpha^3 = \alpha^4 = 1 + \alpha,$$
$$\alpha^2 \cdot \alpha^3 = \alpha^5 = \alpha + \alpha^2,$$
$$\alpha^3 \cdot \alpha^3 = \alpha^6 = \alpha^2 + \alpha^3,$$

то

$$\alpha^3(a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3) = a_0\alpha^3 + a_1(1 + \alpha) + a_2(\alpha + \alpha^2) + a_3(\alpha^2 + \alpha^3) = a_1 + (a_1 + a_2)\alpha + (a_2 + a_3)\alpha^2 + (a_0 + a_3)\alpha^3.$$

Если обозначить через a'_0, a'_1, a'_2, a'_3 соответственно коэффициенты векторного представления a_0, a_1, a_2, a_3 элемента поля $a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3$ после умножения последнего на α^3, то будем иметь



Фиг. 7.12. Схема для вычисления $r(\alpha^3)$.

$$a'_0 = a_1,$$
$$a'_1 = a_1 + a_2,$$
$$a'_2 = a_2 + a_3,$$
$$a'_3 = a_0 + a_3.$$

Эти суммы могут быть вычислены с помощью изображенной на фиг. 7.12 схемы, представляющей собой регистр сдвига с обратной связью. Таким образом, если ввести последовательно в этот регистр символы принятой последовательности r_{n-1}, r_{n-2}, ..., то в результате в регистре окажется $r(\alpha^3)$. При использовании этой схемы вычисление синдрома кончается в тот момент, когда завершается прием слова.

После того как синдром вычислен, следующей операцией при декодировании БЧХ-кодов является нахождение многочлена, корнями которого являются локаторы ошибок. Эта операция является наиболее сложной при технической реализации. При использовании для этой цели как алгоритма Питерсона,

так и предложенного позже алгоритма Берлекэмпа — Месси при большом числе исправляемых ошибок достаточно простого устройства, реализующего эту операцию, построить не удается. В то же время как в том, так и другом случаях математически сложные расчеты могут быть относительно легко выполнены программными методами. Поэтому использование постоянных и полупостоянных памятей и других подобных устройств для построения специализированных вычислительных устройств, предназначенных для декодирования БЧХ-кодов, позволяет снизить стоимость и упростить структуру декодеров.

Однако, когда скорость передачи информации велика, быстродействие универсальной вычислительной машины может оказаться недостаточным для выполнения декодирования поступающих слов. В этих случаях для реализации декодирования часто применяются специализированные устройства. Вопрос о том, какой из алгоритмов — Питерсона или Берлекэмпа — Месси — в этом случае лучше, должен в каждом конкретном случае исследоваться специально.

Как уже указывалось выше, вычисление синдрома осуществляется параллельно с приемом слова и завершается после ввода последнего в буферный регистр. Как будет показано ниже, определение положения ошибок при использовании алгоритма Ченя выполняется при считывании принятого слова из буфера. Таким образом, нахождение многочлена, корнями которого являются локаторы ошибок, должно выполняться независимо от приема переданного слова и считывания последнего из буфера. Поэтому к быстроте выполнения этой операции, а именно времени вычисления многочлена локаторов ошибок, предъявляются повышенные требования.

Если число исправляемых ошибок невелико, коэффициенты многочлена локаторов ошибок $\sigma(X)$ сравнительно просто определить, выразив их через символы синдрома и подставив в полученные формулы конкретные значения символов синдрома. При этом соответствующие схемы оказываются не очень сложными.

Если многочлен

$$\sigma(X) = \sigma_0 + \sigma_1 X + \ \ldots \ + \sigma_t X^t = \sum_{l=0}^{t} \sigma_l X^l, \qquad (7.10)$$

корнями которого являются локаторы ошибок, найден, то непосредственное исправление ошибок можно осуществить, например, воспользовавшись следующим алгоритмом Ченя. Предположим, что при передаче кодового слова БЧХ-кода произошло t ошибок. Тогда, если i-я компонента принятого слова является ошибочной, то

$$\sigma(\alpha^{-i}) = \sum_{l=0}^{t} \sigma_l \alpha^{-li} = 0, \qquad (7.11)$$

Пусть

$$\sigma_{j,\,i} = \sigma_j \alpha^{-ji}. \tag{7.12}$$

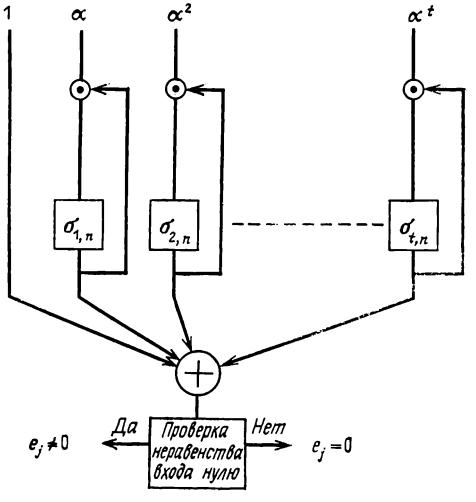
Тогда, как легко видеть, имеют место следующие равенства:

$$\sigma_{j,\,n-1} = \sigma_j \alpha^{-(n-1)j} = \sigma_j \alpha^j \tag{7.13}$$

и

$$\sigma_{j,\,i-1} = \sigma_j \alpha^{-(i-1)j} = \sigma_j \alpha^{-ij} \alpha^j = \sigma_{j,\,i} \alpha^j. \tag{7.14}$$

С помощью схемы, показанной на фиг. 7.13, для каждого i можно вычислить сумму $\sum_{j=0}^{t} \sigma_{j,\,i}$. Если эта сумма равно 0, то, как следует из формулы (7.11), i-я компонента принятого кодо-



Фиг. 7.13. Схема, реализующая алгоритм Ченя.

вого слова содержит ошибку. Сначала в регистр схемы, изображенной на фиг. 7.13, помещаются символы $\sigma_1, \sigma_2, \ldots, \sigma_t$. Далее эти символы умножаются соответственно на $\alpha, \alpha^2, \ldots, \alpha^t$ и проверяется справедливость равенства $e_{n-1}=0$ или равенства $e_{n-1}=1$. Затем содержимое ячеек регистра вновь умножается на $\alpha, \alpha^2, \ldots, \alpha^t$ и находится e_{n-2}. Точно так же находятся и остальные ошибки.

в) Реализация кодов, допускающих мажоритарное декодирование. Как указывалось в предыдущем разделе, техническая реализация БХЧ-кодов, исправляющих большое число ошибок, оказывается сложной. Коды, допускающие мажоритарное декодирование, вообще говоря, имеют несколько худшие корректирующие способности, чем БХЧ-коды, но для них можно по-

строить простые декодирующие устройства. Заметим, что эти коды также являются циклическими. Поэтому будем считать, что в принятой последовательности $r_{n-1}, r_{n-2}, \ldots, r_0$ символы $r_{n-1}, r_{n-2}, \ldots, r_{n-k}$ являются информационными, а символы $r_{n-k-1}, \ldots, r_0$ — проверочными. При декодировании таких кодов удобно вначале вычислять составные проверки, ортогональные относительно шумового символа e_{n-1}, воздействующего на принятый символ r_{n-1}. При этом декодирование происходит следующим образом:

Шаг 1. Вычисляется синдром.

Шаг 2. По $n-k$ символам синдрома строятся J составных проверок, ортогональных относительно e_{n-1}; полученные J составных проверок подаются на входы мажоритарного элемента. Эта операция повторяется L раз, и если выход последнего мажоритарного элемента оказывается равным 1, то декодер принимает решение о том, что в первом символе принятой последовательности содержится ошибка. Если же выход последнего мажоритарного элемента равен 0, то декодер принимает решение о том, что первый символ принятой последовательности является правильным.

Шаг 3. Первый символ принятой последовательности считывается из буфера и складывается по модулю 2 с выходным символом последнего мажоритарного элемента; этим завершается исправление ошибки в первом символе.

Шаг 4. Осуществляется сдвиг синдрома и буферного регистра. На этом шаге необходимо устранить воздействие e_{n-1} на синдром. Если обозначить шумовую последовательность через $E(X)$, сумму членов $E(X)$ степени $n-2$ и менее через $E_{n-2}(X)$, то синдром $S(X)$ можно представить в виде

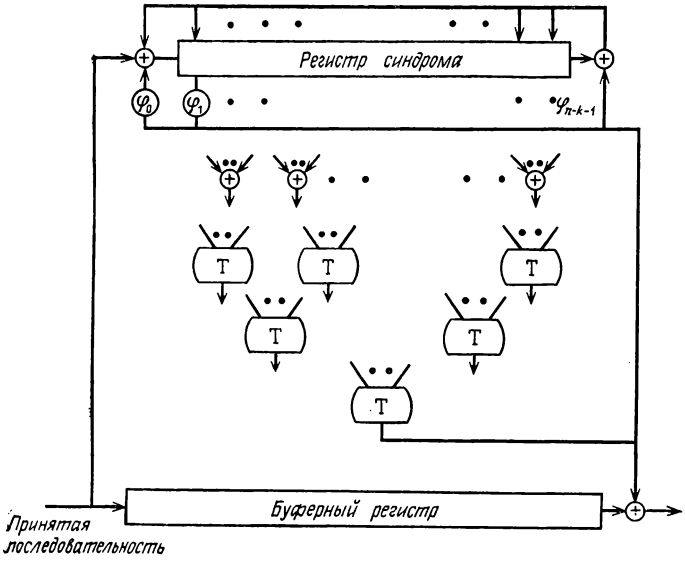
$$S(X) = \mathrm{re}\left[\frac{E(X)}{g(X)}\right] = \mathrm{re}\left[\frac{e_{n-1}X^{n-1}}{g(X)} + \frac{E_{n-2}(X)}{g(X)}\right] \qquad (7.15)$$

(здесь re [ ] — остаток выражения, стоящего в квадратных скобках). Следовательно, воздействие e_{n-1} на синдром можно устранить, сложив синдром с многочленом

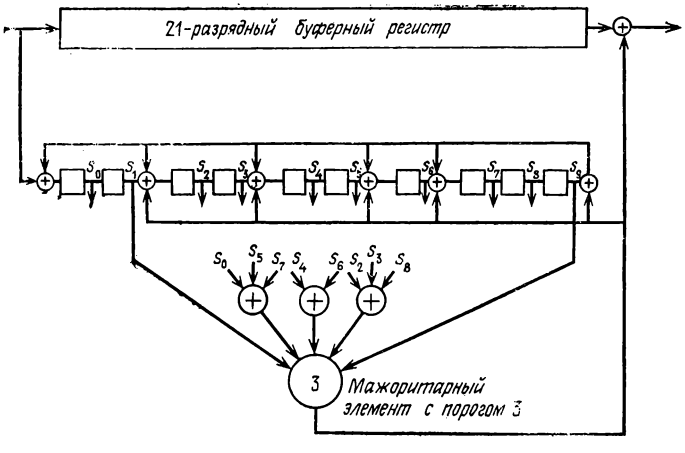
$$\varphi(X) = \varphi_0 + \varphi_1 X + \ldots + \varphi_{n-k-1}X^{n-k-1}, \qquad (7.16)$$

представляющим собой остаток от деления X^{n-1} на $g(X)$. Эту операцию называют коррекцией синдрома. После коррекции синдрома в регистре синдрома содержится синдром принятой последовательности, сдвинутой на один символ.

Шаг 5. Новый синдром, полученный на шаге 4, используется для декодирования 2-го символа принятой последовательности. Вновь выполняя операции 2, 3 и 4, можно декодировать 2-й символ принятой последовательности точно так же, как и 1-й символ.



Фиг. 7.14. Мажоритарный декодер типа I в общем случае.



Фиг. 7.15. Мажоритарный декодер типа I циклического кода с порождающим многочленом $g(X) = 1 + X^2 + X^4 + X^6 + X^7 + X^{10}$; код получается из совершенного разностного множества.

Шаг 6. Точно так же декодер осуществляет декодирование остальных символов принятой последовательности.

Блок-схема декодера, осуществляющего декодирование принятой последовательности описанным выше образом, приведена на фиг. 7.14. Изображенный на фиг. 7.14 декодер называют декодером типа I. На фиг. 7.15 показан декодер типа I циклического кода с порождающим многочленом $g(X)=1+X^2+X^4+X^6+X^7+X^{10}$, получающегося из совершенного разностного множества.

Еще один споособ декодирования рассматриваемых кодов можно получить, построив составные проверки на основе проверочной матрицы кода H. Обозначим здесь через

$$\boldsymbol{h}^{(j)}=(h_0^{(j)}, h_1^{(j)}, \ldots, h_{n-1}^{(j)}) \tag{7.17}$$

элементы пространства строк проверочной матрицы H, а через

$$T=(t_0, t_1, \ldots, t_{n-1}),$$
$$E=(e_0, e_1, \ldots, e_{n-1}),$$
$$R=(r_0, r_1, \ldots, r_{n-1})$$

соответственно передаваемый кодовый вектор, вектор ошибок и принятый вектор. Скалярное произведение R и $\boldsymbol{h}^{(j)}$ имеет вид:

$$R\boldsymbol{h}^{(j)}=(T+E)\boldsymbol{h}^{(j)}=T\boldsymbol{h}^{(j)}+E\boldsymbol{h}^{(j)}. \tag{7.18}$$

Так как $T\boldsymbol{h}^{(j)}=0$, то

$$R\boldsymbol{h}^{(j)}=E\boldsymbol{h}^{(j)}, \tag{7.19}$$

или, что то же самое,

$$h_0^{(j)}r_0+h_1^{(j)}r_1+\ \ldots\ +h_{n-1}^{(j)}r_{n-1}=h_0^{(j)}e_0+ \\ +h_1^{(j)}e_1+\ \ldots\ +h_{n-1}^{(j)}e_{n-1}. \tag{7.20}$$

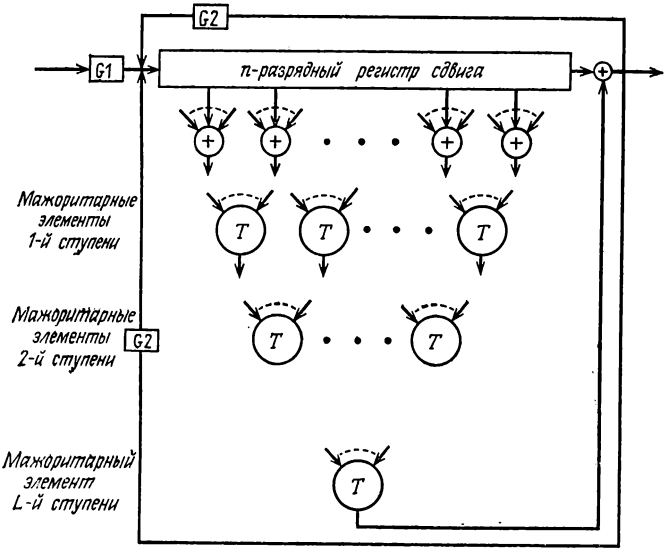
Как видно, отыскание J составных проверок, ортогональных относительно e_{n-1}, эквивалентно отысканию J векторов $\boldsymbol{h}^{(j)}=(h_0^{(j)}, h_1^{(j)}, \ldots, h_{n-1}^{(j)})$ в пространстве строк матрицы H; $(n-1)$-е элементы $h_{n-1}^{(j)}$, $j=1, 2, \ldots, J$, строк равны 1, и строки таковы, что ни при каком $i\neq n-1$ среди элементов $h_i^{(1)}, h_i^{(2)}, \ldots h_i^{(J)}$ нет двух единиц. Согласно формуле (7.15), в качестве составных проверок, ортогональных относительно e_{n-1}, можно взять следующие проверки:

$$A_1=R\boldsymbol{h}^{(1)}=h_0^{(1)}r_0+h_1^{(1)}r_1+\ \ldots\ +h_{n-1}^{(1)}r_{n-1},$$
$$A_2=R\boldsymbol{h}^{(2)}=h_0^{(2)}r_0+h_1^{(2)}r_1+\ \ldots\ +h_{n-1}^{(2)}r_{n-1},$$
$$\vdots$$
$$A_J=R\boldsymbol{h}^{(J)}=h_0^{(J)}r_0+h_1^{(J)}r_1+\ \ldots\ +h_{n-1}^{(J)}r_{n-1}.$$

J составных проверок представляют собой J сумм принятых символов. Векторы $\boldsymbol{h}^{(j)}$ показывают, какие из принятых символов нужно сложить, чтобы получить проверки A_j.

Декодер, осуществляющий декодирование на основе описанных выше составных проверок, называют декодером типа II. Декодер типа II в общем виде изображен на фиг. 7.16 и функционирует следующим образом:

Шаг 1. Вентиль 1 открыт, вентиль 2 закрыт, и принятая последовательность вводится в буферный регистр. После ввода



Фиг. 7.16. Мажоритарный декодер типа II в общем случае.

принятой последовательности вентиль 1 закрывается, а вентиль 2 открывается.

Шаг 2. Строятся составные проверки, ортогональные относительно e_{n-1} или совокупности символов, содержащих e_{n-1}. Для этого определенные принятые символы подаются на входы сумматоров по модулю 2.

Шаг 3. J составных проверок подаются на входы мажоритарных элементов 1-й ступени. Символы с выходов мажоритарных элементов 1-й ступени подаются на входы мажоритарных элементов 2-й ступени и т.д. Символ с выхода мажоритарного элемента L-й ступени складывается по модулю 2 с первым принятым символом r_{n-1}, считываемым из буферного регистра, и таким образом исправление ошибки в r_{n-1} завершается.

Шаг 4. После выполнения шага 3 содержимое буферного регистра сдвигается на один символ вправо. При этом 2-й принятый символ переходит на место 1-го принятого символа и декодируется точно так же, как и 1-й принятый символ.

Шаг 5. Повторяя указанные выше операции 2, 3 и 4, декодер точно так же декодирует остальные принятые символы.

После окончания декодирования всех принятых символов в буферном регистре содержится кодовый вектор и на входы мажоритарных элементов подаются символы 0.

В качестве примера декодера типа II рассмотрим декодер кода над PG (3, 2) с параметрами $n=15$, $k=10$, $t=1$, ортогонализируемого в два шага. Этот код задается нуль-матрицей (проверочной матрицей), в качестве которой берется матрица инцидентности D, строками которой являются в свою очередь всевозможные циклические сдвиги последовательности коэффициентов многочлена [1]

$$w(X)=1+X+X^2+X^4+X^6+X^8+X^{11}. \qquad (7.21)$$

Заметим, что каждой строке матрицы D соответствует одно из 15 проективных пространств PG (2, 2), содержащихся в PG (3, 2):

$$D=\begin{bmatrix}
1&1&1&0&1&1&0&0&1&0&1&0&0&0&0\\
0&1&1&1&0&1&1&0&0&1&0&1&0&0&0\\
0&0&1&1&1&0&1&1&0&0&1&0&1&0&0\\
0&0&0&1&1&1&0&1&1&0&0&1&0&1&0\\
0&0&0&0&1&1&1&0&1&1&0&0&1&0&1\\
1&0&0&0&0&1&1&1&0&1&1&0&0&1&0\\
0&1&0&0&0&0&1&1&1&0&1&1&0&0&1\\
1&0&1&0&0&0&0&1&1&1&0&1&1&0&0\\
0&1&0&1&0&0&0&0&1&1&1&0&1&1&0\\
0&0&1&0&1&0&0&0&0&1&1&1&0&1&1\\
1&0&0&1&0&1&0&0&0&0&1&1&1&0&1\\
1&1&0&0&1&0&1&0&0&0&0&1&1&1&0\\
0&1&1&0&0&1&0&1&0&0&0&0&1&1&1\\
1&0&1&1&0&0&1&0&1&0&0&0&0&1&1\\
1&1&0&1&1&0&0&1&0&1&0&0&0&0&1
\end{bmatrix}. \qquad (7.22)$$

Среди подпространств PG (1, 2) существуют три подпространства $(\alpha^{14}, \alpha^{13}, \alpha^2)$, $(\alpha^{14}, \alpha^{12}, \alpha^5)$ и $(\alpha^{14}, \alpha^{11}, \alpha^{10})$, содержащие элемент α^{14}, соответствующий r_{14}. Следовательно, строками

[1] Матрица D выполняет роль проверочной матрицы кода. Она содержит больше чем $n-k$ строк, но поскольку среди них есть линейно зависимые, то задает код размерности k.

матрицы D, ортогональными относительно $e_{14}+e_{13}+e_2$, являются 10, 13 и 14-я строки. Кроме того, 5, 11 и 13-я строки матрицы D ортогональны относительно $e_{14}+e_{12}+e_5$, а 7, 10 и 11-я строки ортогональны относительно $e_{14}+e_{11}+e_{10}$. Вычисляя следующие три суммы по модулю 2, в каждую из которых входит по 7 символов:

$$r_2+r_4+r_9+r_{10}+r_{11}+r_{13}+r_{14},$$

$$r_1+r_2+r_5+r_7+r_{12}+r_{13}+r_{14},$$

$$r_0+r_2+r_3+r_6+r_8+r_{13}+r_{14}$$

и подавая их на вход порогового элемента с порогом 2, получаем составную проверку, ортогональную относительно $r_2+r_{13}+r_{14}$. Вычисляя суммы

$$r_4+r_5+r_6+r_8+r_9+r_{12}+r_{14},$$

$$r_0+r_3+r_5+r_{10}+r_{11}+r_{12}+r_{14},$$

$$r_1+r_2+r_5+r_7+r_{12}+r_{13}+r_{14},$$

получаем составную проверку, ортогональную относительно $r_5+r_{12}+r_{14}$. Аналогично, вычисляя суммы

$$r_1+r_6+r_7+r_8+r_{10}+r_{11}+r_{14},$$

$$r_2+r_4+r_9+r_{10}+r_{11}+r_{13}+r_{14},$$

$$r_0+r_3+r_5+r_{10}+r_{11}+r_{12}+r_{14},$$

получаем составную проверку, ортогональную относительно $r_{10}+r_{11}+r_{14}$. Если выходы этих трех пороговых элементов подать на вход четвертого порогового элемента, порог которого также равен 2, то на выходе последнего порогового элемента получим проверку, ортогональную относительно e_{14}. Схема декодера, выполняющего описанные выше операции, приведена на фиг. 7.17.

г) Декодирование циклических кодов, исправляющих пачки ошибок. Как циклические коды, так и укороченные циклические коды, предложенные Касами и другими авторами для исправления пачек ошибок, задаются либо порождающим многочленом $g_0(X)$, либо порождающим многочленом $g(X)=g_0(X^i)$, получающимся перемежением $g_0(X)$ с шагом i.

Декодирование таких кодов, исправляющих пачки ошибок, осуществляется следующим образом. Предположим, что корректируемая пачка ошибок длины $b>n-k$ возникла в проверочных символах кодового слова. Пусть $e(X)$ — многочлен, описывающий конфигурацию данной пачки ошибок, а $e_i(X)$ и $e_p(X)$ — многочлены ошибок, соответствующие соответственно

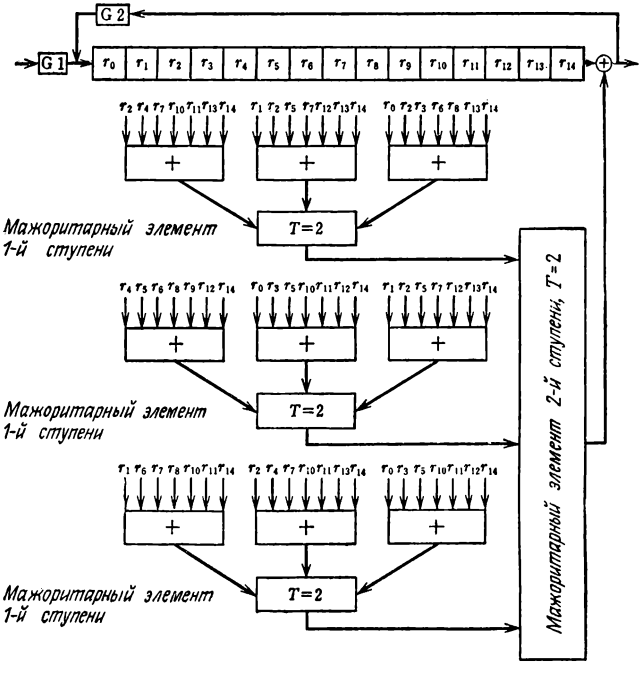
информационным и проверочным символам кодового слова. Выше предполагалось, что

$$e(X) = e_p(X). \tag{7.23}$$

Разделив $e(X)$ на $g(X)$, синдром $S(X)$ можно представить в следующем виде:

$$S(X) = \text{re}\left[\frac{e(X)}{g(X)}\right] = \text{re}\left[\frac{e_p(X)}{g(X)}\right] = e_p(X), \tag{7.24}$$

где re [ ] — остаток, получающийся в результате деления, указанного в квадратных скобках. Таким образом, в данном



Фиг. 7.17. Мажоритарный декодер типа II проективно-геометрического кода.

случае пачка ошибок может быть исправлена путем простого сложения синдрома с принятой последовательностью. Далее рассмотрим случай, когда многочлен $e(X)$, описывающий конфигурацию пачки ошибок, не совпадает с многочленом ошибок $e'(X)$. Предположим, что $e'(X)$ получается i-кратным циклическим сдвигом из $e(X)$, т. е. $e'(X) = X^i e(X)$. Воспользуемся здесь следующей теоремой.

Теорема 7.1. *Синдром последовательности, получающейся i-кратным циклическим сдвигом принятого кодового слова за-*

данной длины n, является i-кратным циклическим сдвигом синдрома принятого кодового слова.

Доказательство. Пусть $s(X)$ — синдром последовательности $e(X)$ длины n. Тогда

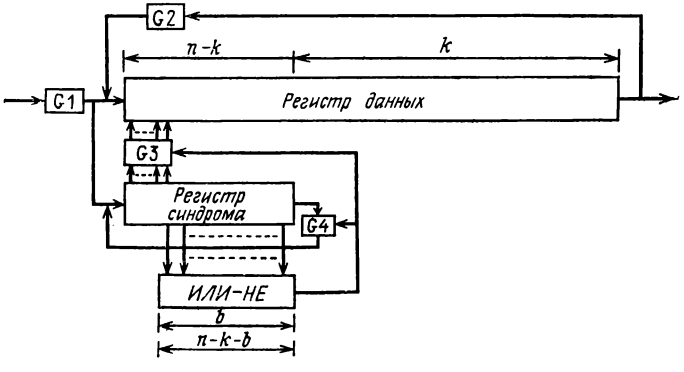
$$s(X) = \mathrm{re}\left[\frac{e(X)}{g(X)}\right], \tag{7.25}$$

$$\frac{e(X)}{g(X)} = q(X) + \frac{s(X)}{g(X)}. \tag{7.26}$$

Пусть $e'(X) = X^i e(X)$ — многочлен, представляющий последовательность, получающуюся в результате i-кратного циклического сдвига $e(X)$ в сторону высших разрядов. Синдром $s(X)$ этого многочлена имеет следующий вид:

$$s'(X) = \mathrm{re}\left[\frac{e'(X)}{g(X)}\right] = \mathrm{re}\left[\frac{X^i e(X)}{g(X)}\right] =$$
$$= \mathrm{re}\left[\frac{X^i g(X)\, q(X) + X^i s(X)}{g(X)}\right] = \mathrm{re}\left[\frac{X^i s(X)}{g(X)}\right]. \tag{7.27}$$

Заметим, что $X^i s(X)$ представляет собой синдром последовательности $e(X)$, циклически сдвинутый на i символов влево.



Фиг. 7.18. Декодер кода, исправляющего пачки ошибок.

Следовательно, если в регистре схемы, предназначенной для вычисления синдрома, получен синдром $s(X)$, то для нахождения $s'(X)$ достаточно просто циклически сдвинуть содержимое этого регистра на i символов.

Таким образом, как следует из этой теоремы, для исправления пачки ошибок $e'(X) = X^i e(X)$ синдром, содержащийся в регистре генератора синдрома, необходимо циклически сдвинуть на i символов и далее сложить с циклически сдвинутой на столько же символов принятой последовательностью.

Декодер кода, исправляющего пачки ошибок описанным выше образом, показан на фиг. 7.18. Элемент ИЛИ — НЕ этого декодера имеет $n-k-b$ входов, на которые подаются символы с выходов регистра синдрома. Если все эти символы равны нулю, то это означает, что b младших разрядов синдрома представляют собой пачку ошибок длины b или менее, которая может быть исправлена. В этом случае цепь обратной связи регистра синдрома с помощью вентиля $G4$ разрывается, принятая последовательность, сдвинутая циклически на нужное число символов, складывается с синдромом, и таким образом пачка ошибок оказывается исправленной.

Для примера рассмотрим код длины $n=15$ с $k=9$ информационными символами, исправляющий пачки ошибок длины 3 и менее. Согласно табл. 7.1, этот код имеет следующий порождающий многочлен:

$$g_0(X) = X^6 + X^5 + X^4 + X^3 + 1.$$

Для исправления пачек ошибок длины 9 и менее этот код следует «проредить» с шагом 3. В этом случае код будет задаваться порождающим многочленом

$$g(X) = X^{18} + X^{15} + X^{12} + X^9 + 1.$$

Рассмотрим код, задаваемый многочленом $g_0(X)$. Предположим, что передавалось нулевое кодовое слово и при передаче возникли ошибки в 0, 1 и 2-м символах. В этом случае

$$e(X) = 1 + X + X^2,$$

а синдром

$$s(X) = \mathrm{re}\,[(1 + X + X^2)/(1 + X^3 + X^4 + X^5 + X^6)] = 1 + X + X^2.$$

Далее предположим, что при передаче возникла пачка ошибок длины 3 в 5, 6 и 7-м символах. В этом случае

$$e'(X) = X^5 + X^6 + X^7 = X^5(1 + X + X^2),$$

а синдром

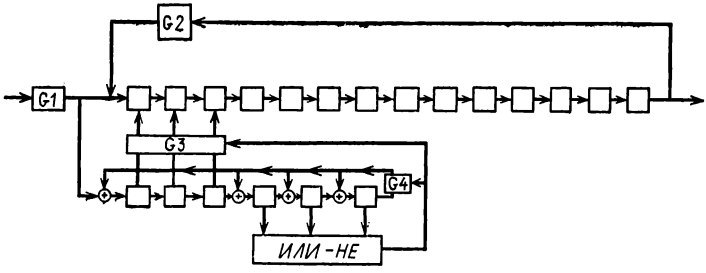
$$s'(X) = \mathrm{re}\,[(X^5 + X^6 + X^7)/(1 + X^3 + X^4 + X^5 + X^6)] = X + X^4.$$

Если этот синдром сдвигать влево, в сторону младших разрядов, то содержимое регистра будет изменяться следующим

образом:

Число сдвигов	Содержимое регистра
0	$X + X^4$
1	$1 + X^3 = X^4 + X^5 + X^6$
2	$X^3 + X^4 + X^5$
3	$X^2 + X^3 + X^4$
4	$X + X^2 + X^3$
5	$1 + X + X^2$

После 5-кратного сдвига старшие $15 - 9 - 3 = 3$ разряда синдрома оказываются равными нулю, а в младших разрядах формируется вектор ошибок. Следовательно, сдвигая принятую



Фиг. 7.19. Декодер кода с порождающим многочленом $g(X) = 1 + X^3 + X^4 + X^5 + X^6$.

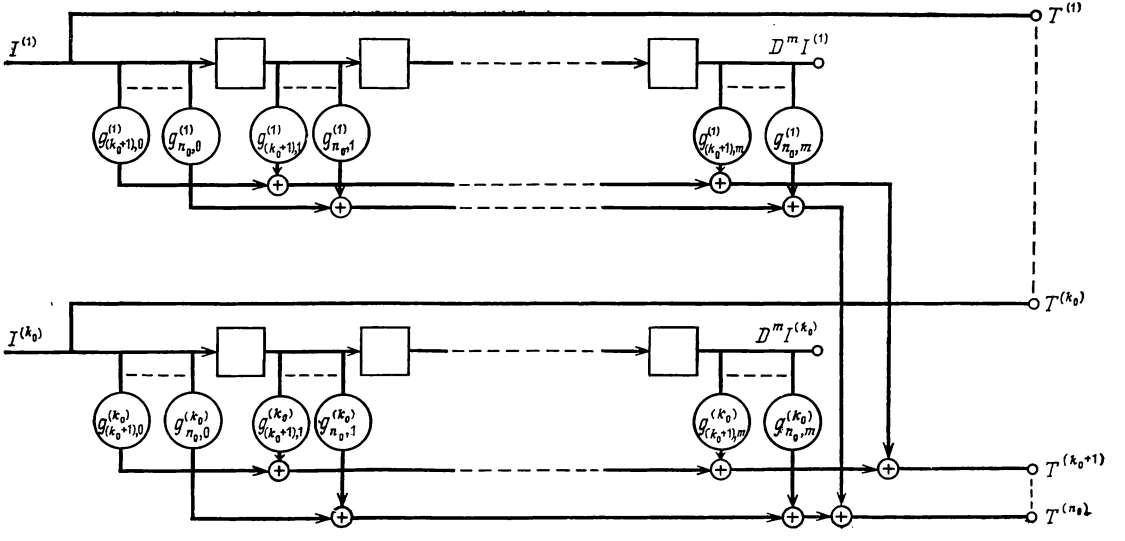
последовательность циклически на 5 символов в сторону младших разрядов и складывая ее по модулю 2 с содержимым регистра, предназначенным для хранения синдрома, можно исправить возникшую пачку ошибок. Декодер, выполняющий эти операции, показан на фиг. 7.19.

7.2. Реализация порогового декодирования

Кодеры и декодеры кодов, допускающих пороговое декодирование, могут быть построены из нескольких стандартных схем.

7.2.1. Кодеры

Имеются два типа кодеров: Rn_A-разрядные кодеры, предложенные Возенкрафтом и Рейффеном [16], $(1-R)n_A$-разрядные кодеры, предложенные Месси [2]. Схема Rn_A-разрядного кодера приведена на фиг. 7.20. В соответствии с формулами (5.5) и (5.6) k_0 информационных последовательностей $I^{(j)}(D)$,



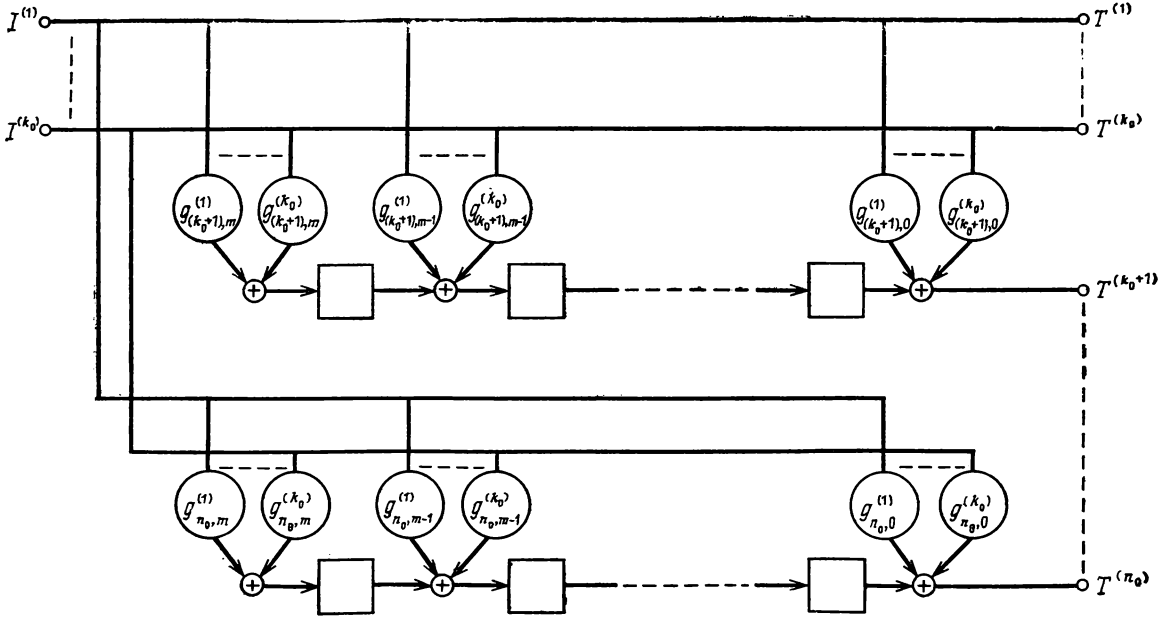
Фиг. 7.20. Rn_A-разрядный кодер [22].

$j=1, \ldots, k_0$, подаются на k_0 входов m-разрядных регистров сдвига и одновременно на выходы кодера для передачи по каналу. Построение $n_0 - k_0$ проверочных последовательностей $T^{(j)}(D)$, $j=k_0+1, \ldots, n_0$, осуществляется с помощью сумматоров по модулю 2 по порождающим многочленам кода.

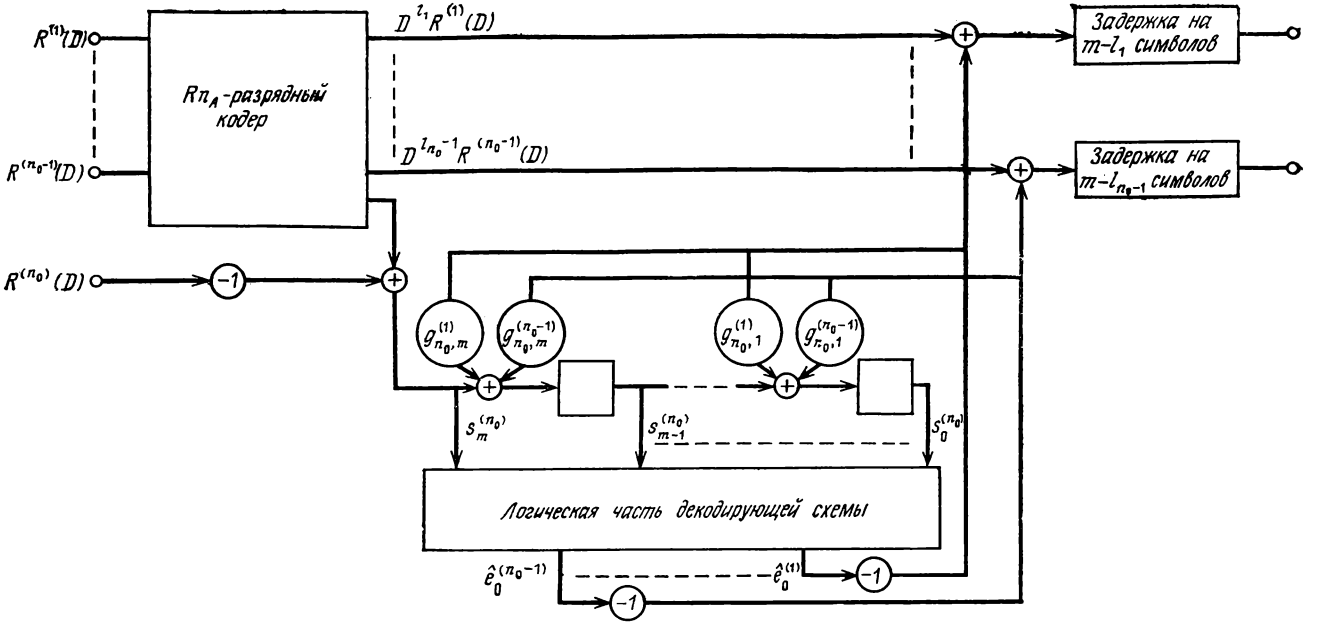
Покажем, что проверочные последовательности на выходе этого кодера действительно порождаются в соответствии с равенством (5.6). Для этого сначала предположим, что $i_0^{(1)}=1$, а все остальные входные символы равны нулю. В этом случае, как легко убедиться, i-я выходная последовательность будет совпадать с $G_{(i)}^{(1)}(D)$ $(i=k_0+1, \ldots, n_0)$. Аналогично можно убедиться, что при поступлении символа 1 на любой другой j-й, $j\neq 1$, вход кодера и символов 0 на остальные входы i-я выходная последовательность будет совпадать с порождающим многочленом $G_{(i)}^{(j)}(D)$ используемого кода. В силу линейности кодера при любом входе последовательности на выходе кодера будут определяться равенством (5.6). Поскольку для построения описанной схемы требуется k_0 регистров сдвига, каждый из которых должен иметь m разрядов, то общее число разрядов регистров в схеме равно $mk_0=R(n_A-n_0)\simeq Rn_A$. Поэтому эту схему называют Rn_A-разрядным кодером.

Другой кодер называется $(1-R)n_A$-разрядным кодером и показан на фиг. 7.21. В этом кодере сумматоры устанавливаются между ячейками памяти регистра сдвига и, следовательно, каждый из них имеет не более $k+1$ входов. Определив отклики схемы на единичные воздействия и воспользовавшись свойством линейности точно так же, как и в случае Rn_A-разрядного кодера, можно проверить, что последовательности на выходе $(1-R)n_A$-разрядного кодера определяются равенствами (5.5) и (5.6). Последний кодер содержит m-разрядный регистр сдвига для каждой из n_0-k_0 проверочных последовательностей и, следовательно, требует для построения $m(n_0-k_0)=(1-R)(n_A-n_0)\simeq(1-R)n_A$ разрядов регистров сдвига.

Если сравнивать описанные выше кодеры по числу разрядов регистров сдвига, необходимых для их построения, то, как легко видеть, при $R>1/2$ лучше иметь $(1-R)n_A$-разрядный кодер, а при $R<1/2$ лучше иметь Rn_A-разрядный кодер. При $R=1/2$ оба эти кодера требуют одно и то же число разрядов регистра сдвига. Однако здесь следует заметить, что в Rn_A-разрядном кодере информационные символы после поступления хранятся еще m интервалов времени без изменений, в частности, этот кодер имеет выходы $D^mI^{(1)}, \ldots, D^mI^{(k_0)}$, тогда как в $(1-R)n_A$-разрядном кодере такие выходы отсутствуют. По этой причине в описанном ниже декодере используется Rn_A-разрядный, а не $(1-R)n_A$-разрядный кодер.



Фиг. 7.21. $(1 - R)\, n_A$-разрядный кодер [22].



Фиг. 7.22. Декодер.

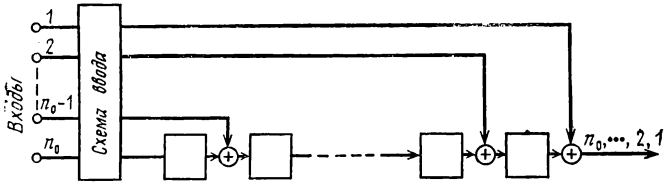
7.2.2. Декодер

Схема декодера приведена на фиг. 7.22, где для простоты $k_0 = n - 1$. Принимаемые информационные последовательности подаются на входы содержащегося в декодере Rn_A-разрядного кодера, который точно так же, как и при кодировании, вычисляет для поступающих на его входы последовательностей проверочную последовательность. Эта проверочная последовательность складывается с принимаемой проверочной последовательностью и таким образом формируется синдром $s_0^{(n_0)}, \ldots, s_m^{(n_0)}$. Синдром вводится в m-разрядный регистр сдвига, с $m+1$ выходов которого в свою очередь символы поступают на входы логической части декодирующей схемы. При использовании самоортогональных кодов логическая часть декодирующей схемы представляет собой совокупность пороговых элементов с порогом $(J/2)+1$. В случае же использования ортогоанализируемых кодов логические части декодирующей схемы, кроме пороговых элементов, содержат также сумматоры по модулю 2, предназначенные для линейного преобразования синдрома. На выходах логической части декодирующей схемы формируются оценки $\hat{e}_0^{(1)}, \ldots, \hat{e}_0^{(n_0-1)}$ шумовых символов $e_0^{(1)}, \ldots, e_0^{(n_0-1)}$. Эти оценки складываются с соответствующими выходными символами Rn_A-разрядного кодера, и таким образом процесс исправления ошибок завершается. После устранения с помощью цепи обратной связи воздействия символов $\hat{e}_0^{(1)}, \ldots, \hat{e}_0^{(n_0-1)}$ на синдром в последующие моменты времени 1, 2, ... точно так же осуществляется исправление ошибок $e_1^{(1)}, \ldots, e_1^{(n_0-1)}, e_2^{(1)}, \ldots$ $\ldots, e_2^{(n_0-1)}, \ldots$.

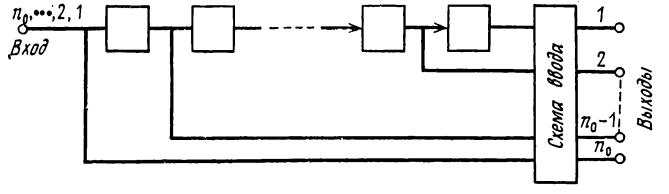
7.2.3. Схема преобразования параллельных слов в последовательные и обратная схема

Если канал один и передача n символов, формирующихся на выходе кодера в каждый момент времени, осуществляется последовательно, то на передающем конце используется схема преобразования параллельных слов в последовательные, а на приемном конце — схема преобразования последовательных слов в параллельные. Эти схемы показаны соответственно на фиг. 7.23 и 7.24. Рассмотрим функционирование схемы преобразования параллельных слов в последовательные, предположив, что передача по каналу n символов осуществляется за время T. Схема ввода соединяет одновременно n_0 своих входов с соответствующими выходами на время T/n_0, и в течение этого времени, как показано на фиг. 7.23, входные символы вводятся в ячейки регистра сдвига. После этого входы и выходы схемы

ввода разъединяются и при последующих сдвигах регистра на выходе последнего последовательно появляются n входных символов. Аналогично осуществляется преобразование последовательных слов в параллельные.



Фиг. 7.23. Схема преобразования параллельных слов в последовательные.



Фиг. 7.24. Схема преобразования последовательных слов в параллельные.

7.2.4. Реализация сверточных кодов, исправляющих пачки ошибок

При реализации сверточных кодов со скоростями передачи $R \geqslant 1/2$, исправляющих пачки ошибок, при кодировании целесообразно использовать описанный выше $(1-R)n_A$-разрядный кодер. Для построения декодера используются Rn_A-разрядный кодер и логическая часть декодирующей схемы. Логическая часть декодирующей схемы, на вход которой подается синдром, формирует (n_0-1) простейших (единичных) конфигураций. Эта схема состоит из n_0-1 элементов И, n_0-2 элементов НЕ и регистра сдвига. В частности, в двоичном случае единичные конфигурации длины k представляют двоичные последовательности длины k, которые обнаруживаются с помощью элемента И, показанного на фиг. 7.25. Из n_0-1 таких элементов И и n_0-2 элементов НЕ, устанавливаемых на выходах элементов И, строится любая логическая часть декодирующей схемы (фиг. 7.26).

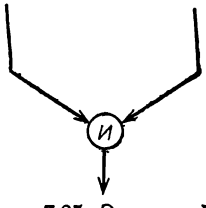
Элемент И, выходные символы которого используются для исправления ошибок в j-й информационной последовательности, будем считать j-элементом И. Пусть B — общий вход всех элементов И, C_j — второй вход j-го элемента И и D_j — выход

последнего. Выход D_j называется функцией исправления ошибок и определяется следующим образом:

$$D_j = BC_j\bar{C}_{j+1} \ldots \bar{C}_{n_0-1}, \tag{7.28}$$

где BC_j соответствует единичной конфигурации j-й принимаемой последовательности, а черта над символами $\bar{C}_{j+1}, \ldots, \bar{C}_{n_0-1}$ означает инверсию последних. Число элементов задержки, показанных на фиг. 7.26, равно β для кодов типа I и $b+n_0-1$ для кодов типа II.

Кодер и декодер кода, рассмотренного в примере 5.10, показаны соответственно на фиг. 7.27 и 7.28. Функции исправле-



Фиг. 7.25. Элемент И.

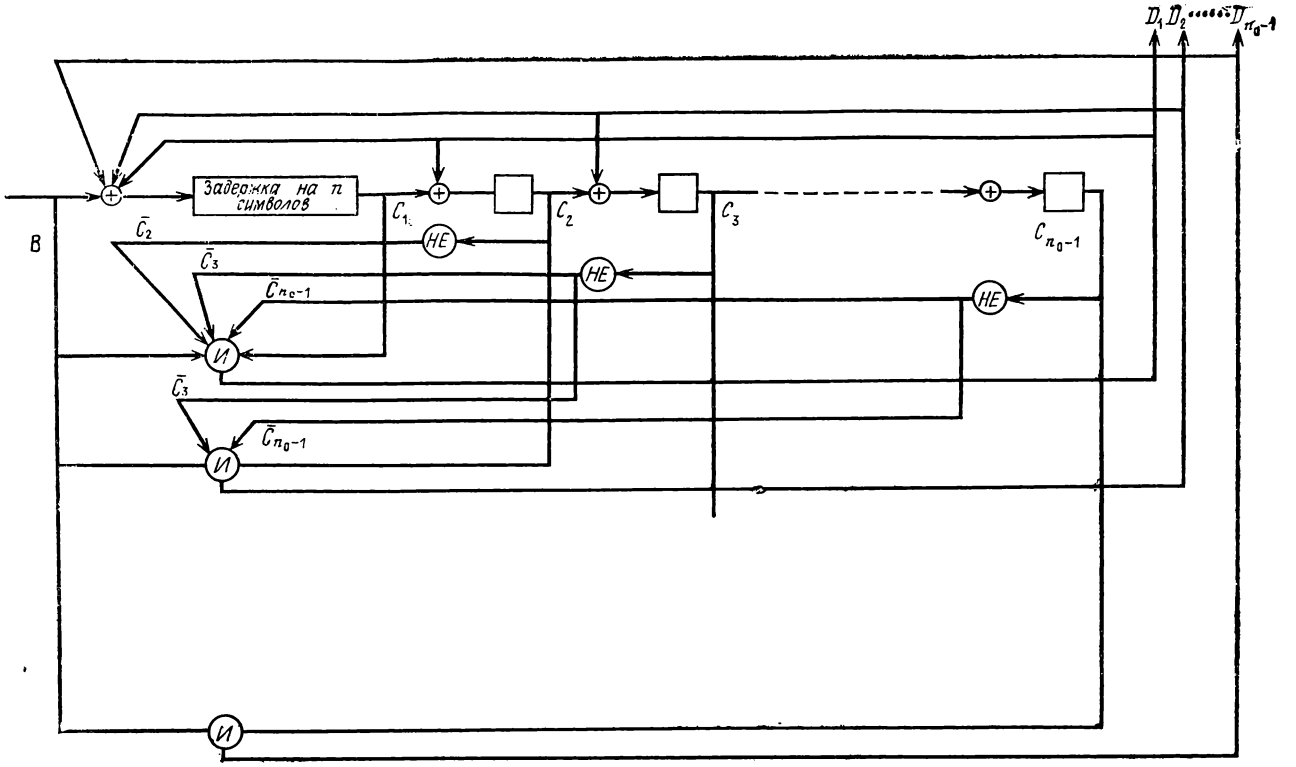
ния ошибок в этом случае определяются равенствами

$$D_1 = BC_1\bar{C}_2,$$
$$D_2 = BC_2.$$

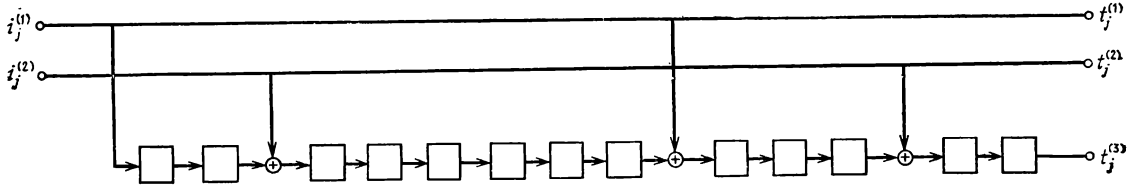
Если синдром содержит символ $e_0^{(2)}$ и $e_0^{(2)} = e_1^{(2)} = 1$, то тем не менее $C_1=0$ и $C_2=1$ (так как $A=B_1=B_2=1$) и ошибка $e_0^{(2)}$ может быть исправлена.

7.3. Обсуждение связи теории кодирования с реальными техническими проблемами

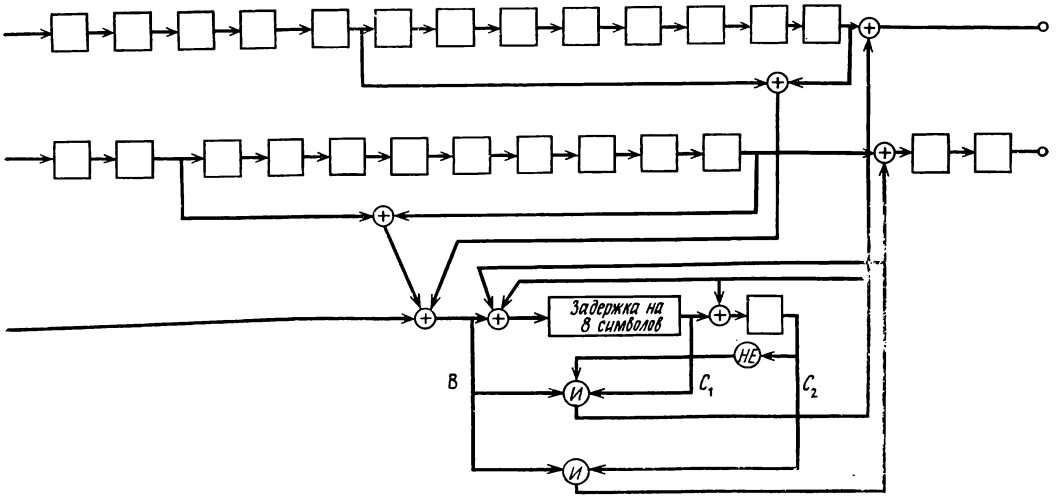
В первых шести главах рассматривались различные классы кодов, исправляющих ошибки. Эта часть книги носила математический, главным образом алгебраический характер. Данная глава посвящена применениям результатов теории кодирования, в частности применениям кодов, исправляющих ошибки. Здесь математические алгоритмы рассматриваются под другим углом зрения, а именно описываются их применения в реальных условиях и те стороны, которые не всегда удается достаточно полно учесть в теории. Ниже рассматриваются вопросы стоимости и повышения надежности при использовании кодирования, различные принятые в теории кодирования условности и другие вопросы.



Фиг. 7.26. Логическая часть декодирующей схемы.



Фиг. 7.27. Кодер кода с $n_0 = 3$ и $\beta = 2$.



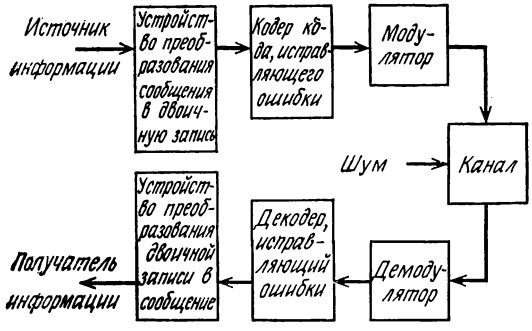
Фиг. 7.28. Декодер кода с $n_0 = 3$ и $\beta = 2$.

7.3.1. Задача теории кодирования

В теории кодирования изучаются и предлагаются методы борьбы с ошибками при обработке информации на ЭВМ и методы помехоустойчивой передачи информации по каналам с шумами, т. е. задача этой теории — повышение надежности работы вычислительных систем и систем связи. В теории кодирования эта задача решается путем введения избыточности в передаваемую информацию или вычислительные устройства. Методы повышения надежности путем введения избыточности широко используются как при повседневной деятельности людей, так и в различных технических устройствах. Например, в английском, японском и других естественных языках всегда имеется избыточность, обусловленная грамматикой, а также структурой слов и предложений данного языка. Используя эту избыточность, мы решаем кроссворды, исправляем ошибки и т. д. Известно, например, что избыточность английского языка достигает 70%. Точно так же в технике устройства могут дублироваться, и при выходе одного из них из строя система переключается на другое. В других случаях используется одно устройство, но каждая операция с помощью этого устройства выполняется дважды, и если результат операции в обоих случаях один и тот же, то считается, что этот результат правильный. Если же результаты различны, то это означает, что произошла ошибка и для ее исправления операция выполняется еще один раз. Во всех этих случаях наличие избыточности позволяет обнаруживать и даже исправлять ошибки и, следовательно, повышает надежность систем связи или вычислительных систем. Поскольку теория кодирования применяется главным образом при построении систем связи и вычислительных систем, то ниже мы ограничимся только их рассмотрением.

Сначала остановимся на системах связи, хотя в настоящее время теория кодирования в этих системах достаточно широко еще не применяется. Это связано с тем, что до настоящего времени, как правило, системы связи проектировались таким образом, чтобы можно было достичь нужной степени надежности без применения кодов, исправляющих ошибки. Кроме того, в случае передачи речи и изображений при наличии небольших шумов качество передачи хотя и снижается, но остается приемлемым для получателя. Отдельные ошибки и искажения получатель, кроме того, может исправить, используя естественную избыточность, содержащуюся в информации такого типа. В то же время в тех случаях, когда естественная избыточность, вводимая источником информации, невелика, например в системах передачи

данных, для достижения нужной достоверности передачи необходимо вводить избыточность в передаваемую информацию искусственно методами теории кодирования. Именно в подобных системах и находят широкое применение методы теории кодирования. Методы теории кодирования применяются главным образом в системах связи двух-трех типов, но по мере роста возможностей разработчиков, увеличения числа видов передаваемой информации и типов систем связи, а также снижения стоимости интегральных и больших интегральных схем область применения теории кодирования постоянно будет расширяться и методы помехоустойчивого кодирования, используемые в настоящее время, найдут применение в системах связи также и других типов.

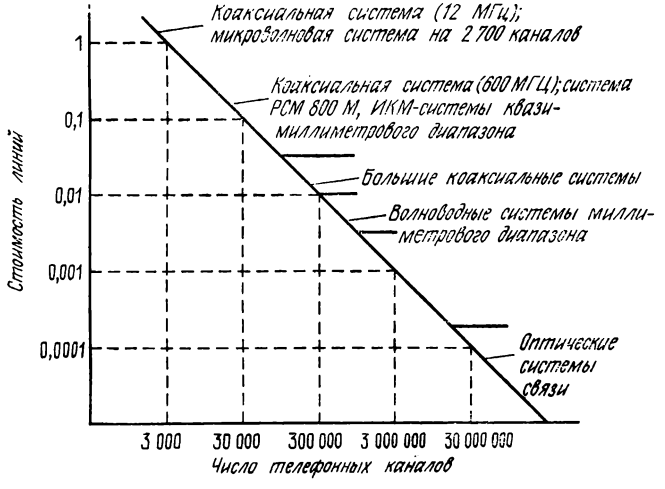


Фиг. 7.29. Модель системы связи.

На фиг. 7.29 показана простейшая модель системы связи, в которой используются коды, исправляющие ошибки. Информация, поступающая от источника информации, с помощью преобразователя «информация → двоичная последовательность» преобразуется в двоичную последовательность, которая направляется на вход кодера некоторого кода, исправляющего ошибки. Двоичная последовательность с выхода этого кодера, в которую введена избыточность, с помощью модулятора преобразуется в сигналы канала и передается по нему. При передаче по каналу на сигналы канала воздействуют шумы, которые искажают их. На приемном конце искаженные шумом сигналы поступают на вход демодулятора, который преобразует их в двоичную последовательность, часть символов которой является избыточной. Декодер, используя эту избыточность, осуществляет исправление ошибок и восстанавливает исходную двоичную информационную последовательность. Эта информационная последовательность преобразуется к виду, удобному для получателя информации, и направляется к нему.

7.3.2. Стоимость и надежность

Рассмотрим взаимосвязь между стоимостью и надежностью систем, использующих помехоустойчивое кодирование. На фиг. 7.30 показана связь между стоимостью и пропускной способностью различных систем связи. Как видно из фиг. 7.30, стоимость одного канала зависит от того, является ли система связи коаксиальной, микроволновой, системой миллиметрового диапазона и т. д., и снижается по мере роста суммарной пропускной способности системы связи. Что касается каждой конкретной системы, то ее стоимость возрастает из года в год только из-за



Фиг. 7.30. Стоимость линий связи.

повышения расходов на оплату обслуживающего персонала. Если же рассматривать стоимость самой системы связи, то она остается постоянной. Повысить надежность данной конкретной системы можно, используя те или иные вспомогательные средства (например, путем введения избыточности). При этом всегда требуется, чтобы стоимость вспомогательных средств существенно не повышала стоимость самой системы связи. Как видно из фиг. 7.30, по мере возрастания пропускной способности системы, использующей вспомогательные средства, относительная стоимость последних уменьшается.

Рассмотрим повышение стоимости системы при введении избыточности. Так как помехоустойчивое кодирование применяется в цифровых системах связи, то различные схемы, необходимые для введения и использования избыточности, строятся из цифровых устройств. Такими устройствами являются, например, полупроводниковые запоминающие устройства, предполагаемое

изменение стоимости которых показано на фиг. 7.30. Согласно прогнозам на ближайшие годы, стоимость цифровых устройств будет по-прежнему быстро снижаться. Как следствие снижения стоимости цифровых устройств будет снижаться стоимость схем, используемых для введения и использования избыточности; в свою очередь это приведет к расширению области применения помехоустойчивого кодирования.

7.3.3. Применение в вычислительных системах

Остановимся здесь на применении теории кодирования в вычислительных системах. Первым кодом, исправляющим ошибки, который стал применяться в вычислительных машинах, был разработанный еще на заре теории кодирования в 1950 г. код Хэмминга. Как указывалось ранее, этот код был разработан специально для применения в вычислительных машинах. Код Хэмминга стал применяться в вычислительных машинах не сразу, и это объясняется тем, что после появления кода Хэмминга постоянно совершенствовались сами запоминающие элементы; сначала использовались электронные лампы, далее появились параметроны, полупроводниковые элементы и наконец интегральные схемы. При этом надежность запоминающих элементов постоянно и быстро росла. Первой вычислительной машиной, в которой использовался код Хэмминга, была вычислительная машина IBM 7030, а в Японии — машина DIPS фирмы Japan Telephone and Telegraph Public. Однако если первая была построена спустя 10 лет после появления кода Хэмминга, то вторая — спустя 20 лет. До этого времени в вычислительных машинах использовался лишь простейший способ повышения надежности, а именно проверка на четность (или на нечетность).

В последние годы различные достаточно сложные коды, исправляющие большое число ошибок, начали применяться как в самих вычислительных машинах, так и в их периферийном оборудовании, в частности в накопителях на магнитной ленте. По мере развития вычислительной техники коды, исправляющие ошибки, будут играть все большую роль в повышении надежности вычислительных систем [1]). Согласно Чену [2], для этого имеются следующие предпосылки:

I. При повышении скорости вычислений и плотности монтажа уменьшается отношение сигнал/шум.

II. Целесообразность использования с экономической точки зрения несовершенных элементов, в частности больших интегральных схем.

[1]) Обзор достижений в области надежных вычислений и надежного хранения информации читатель может найти в статье Б. С. Цыбакова, С. И. Гельфанда, А. В. Кузнецова, С. М. Артюкова «Reliable computation and reliable storage of information», Proc. of 1975 IEEE — USSR Joint Werkshop on Inform. Theory, IEEE Press, N. Y., 1976, p. 216. — *Прим. ред.*

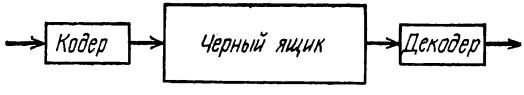
III. По мере увеличения числа вычислительных машин, использующих каналы связи, т. е. по мере расширения сетей вычислительных машин, возрастает объем информации, передаваемой в режиме «on line».

Указанные выше закономерности связаны между собой. Так, например, существует хорошо известный закон Гроша [3], устанавливающий связь между стоимостью и скоростью обработки информации в вычислительных машинах. Согласно этому закону, стоимость вычислений пропорциональна корню квадратному из скорости вычислений. Так, например, для снижения стоимости вычислений в 10 раз скорость вычислений должна быть увеличена в 100 раз. Однако если повышается скорость вычислений, то уменьшаются отношение сигнал/шум и, естественно, надежность. В этом случае обычно возникает необходимость вновь повысить надежность путем введения избыточности, которая существенно не повлияла бы на стоимость системы в целом. Еще одно обстоятельство, связанное с предпосылкой II, на которое следует обратить внимание, — это то, что в качестве оперативной памяти современных вычислительных машин начали использоваться полупроводниковые запоминающие устройства, стоимость которых, как указывалось, быстро снижается.

Для сравнения надежности полупроводниковых запоминающих устройств и запоминающих устройств на магнитных элементах рассмотрим, например, ЗУ средней емкости, состоящие из накопителей и периферийного оборудования. В ЗУ на магнитных элементах доля отказов из-за неисправности накопителя составляет 10—40%, а доля отказов из-за неисправности периферийных схем — от 60 до 90%. В то же время в полупроводниковых ЗУ доля отказов из-за неисправности накопителя достигает 70—95% и лишь 5—30% составляют отказы периферийного оборудования [4]. Таким образом, при использовании полупроводниковых ЗУ в качестве оперативной памяти возникает необходимость повышения надежности последней. Главным образом именно в связи с этим в настоящее время обсуждаются различные способы повышения надежности оперативной памяти с помощью кодов, исправляющих ошибки. Конечно, введение избыточности приводит к повышению стоимости памяти периферийного оборудования, но если учесть, что это повышение позволяет резко понизить стоимость накопителей, то в целом стоимость вычислений понижается. С точки зрения возможности понижения стоимости вычислений применение кодов, исправляющих ошибки, в вычислительных машинах также представляется целесообразным.

Выше главным образом рассматривалось применение в вычислительных системах кодов, исправляющих ошибки. Однако для повышения надежности вычислительных систем используют-

ся также «самокорректирующиеся» схемы, изучаемые в теории автоматов, а также так называемые арифметические коды, описываемые в последующих двух главах и используемые в некоторых арифметических устройствах, в частности в сумматорах. Что касается кодов, исправляющих ошибки, то они в общем случае применяются в системах, про которые достаточно знать лишь то, что в них информация вводится и выводится. При этом исследуемая система часто оказывается сложнее, чем изображенный на фиг. 7.29 канал связи. В действительности, как показано на фиг. 7.31, любую систему можно рассматривать как некий «черный ящик» и, сравнивая входы с выходами, найти распределение ошибок. Зная это распределение, из всех известных кодов можно выбрать тот, который исправляет наиболее часто возникающие конфигурации ошибок; кроме того, можно построить новый код специально для данного распределения ошибок. Например, в процессе общения оператора с вычислительной машиной при выполнении ручных операций возникает большое число



Фиг. 7.31. Система связи с каналом, рассматриваемым как «черный ящик».

ошибок, связанных с перестановкой соседних символов. Эти ошибки представляют собой специальные пачки ошибок длины 2, и для их исправления могут быть разработаны специальные коды, учитывающие этот характер ошибок.

Для определения ухудшения качества и эффективности передачи существующих систем связи широко используются аналитические методы, с помощью которых разрабатываются методы улучшения характеристик систем связи. Что касается диалога человека с машиной, то для определения причин возникновения ошибок здесь необходимо анализировать сложные процессы, протекающие от клеток мозга до нервных окончаний человека, что, конечно, практически сделать невозможно. Для борьбы с ошибками оператора можно использовать алгебраические коды, рассматривая при этом тракт ввода как «черный ящик» и просто устанавливая на его входе кодер, а на выходе — декодер.

7.4. Различные предположения, используемые в теории кодирования

Рассмотрим здесь различные предположения, используемые в теории кодирования. В теории кодирования применяются главным образом алгебраические методы, в частности теория Галуа,

и это позволило достаточно быстро достичь значительных успехов. При этом некоторые проблемы, для решения которых алгебраические методы применить было трудно, остались в значительной степени нерешенными. Возможность решения алгебраическими методами одних проблем и трудности решения других не менее важных проблем оказали существенное влияние на формирование теории кодирования. Для того чтобы можно было применить алгебраические методы, в теории кодирования вводились различные предположения, что ограничивало область применения полученных результатов. Знание этих предположений очень важно для понимания взаимосвязи теории кодирования с теми реальными проблемами, которые она призвана решать.

7.4.1. Дискретные величины

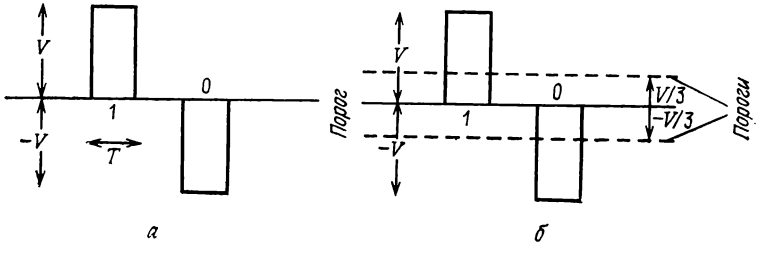
В теории кодирования подлежащая передаче, хранению или какой-либо обработке информация обычно представляется в виде последовательностей целых чисел, в частности в простейшем случае в виде последовательностей символов 0 и 1. Поэтому при изучении передачи информации в теории кодирования рассматриваются лишь дискретные каналы. Чтобы коды, исправляющие ошибки, можно было применить в реальных непрерывных каналах, непрерывные величины с помощью аналого-цифровых преобразователей должны преобразовываться в дискретные величины.

7.4.2. Методы принятия решений

Предположим, что для передачи символов 0 и 1 используются прямоугольные импульсы длительности T единиц времени с амплитудами соответственно $-V$ и V. При передаче по каналу с аддитивным гауссовским шумом эти символы искажаются и на вход приемника поступают символы, имеющие случайную амплитуду. В качестве решающего элемента при приеме может использоваться пороговый элемент с порогом 0, который классифицирует принятый импульс как 0, если его амплитуда меньше нуля, и как 1, если амплитуда больше нуля. Такой способ принятия решения о переданном символе называется *жестким*. Кроме того, существует также так называемый *мягкий* способ принятия решений, когда решающий элемент имеет два порога $V/3$ и $-V/3$, как показано на фиг. 7.32, *б*. При этом выход решающего элемента принимает значение 1, если амплитуда принятого импульса больше $V/3$, значение 0, если амплитуда принятого импульса меньше $-V/3$, и решающий элемент не принимает никакого решения относительно 0 и 1, если амплитуда импульса находится между $V/3$ и $-V/3$. При этом, кроме ошибок типа пере-

ход символа 1 в символ 0 или наоборот, могут возникать также так называемые стирания, когда решающий элемент решения о переданном символе не принимает. В этом случае на выходе решающего элемента формируется так называемый символ стирания. При передаче последовательности символов место возникновения ошибок типа переход символа 1 в 0 или наоборот априори неизвестно. В то же время положение стираний при приеме известно, а поэтому исправить их оказывается значительно проще, чем исправить ошибки. Поскольку в канале с аддитивным гауссовским шумом стираний возникает значительно больше, чем ошибок, то путем использования мягкого способа принятия решений можно упростить процесс исправления ошибок в целом.

Далее рассмотрим случай, когда описанные выше импульсы передаются по каналу блоками по K импульсов в каждом, но о

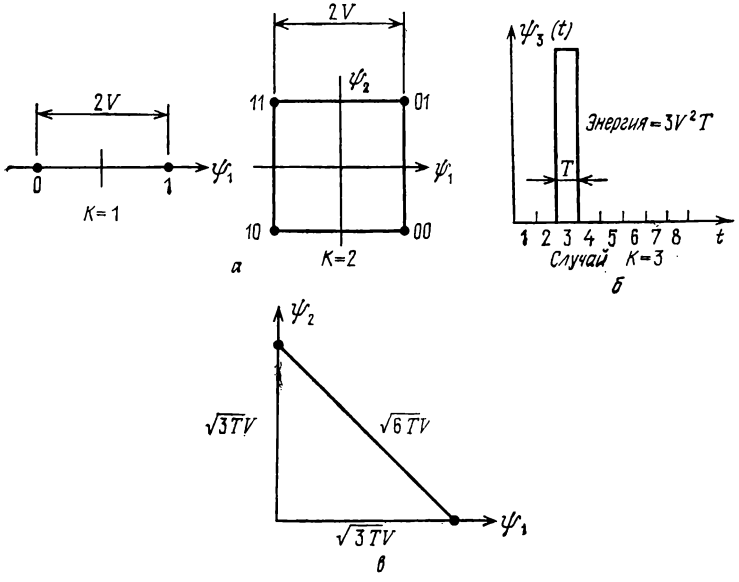


Фиг. 7.32. Импульсы 1 и 0 и значения порогов.

каждом отдельном переданном символе принимается жесткое решение, не зависящее от других принятых импульсов. Предположим, что шум в канале является аддитивным и гауссовским. Обозначим через p_e вероятность возникновения ошибки в отдельном символе. Тогда вероятность того, что в принятом блоке нет ни одной ошибки, равна $(1-p_e)^K$ и убывает с ростом K. Это связано с тем, что при увеличении K расстояние между «соседними» сигналами остается постоянным, равным $2V$ (фиг. 7.33, *а*), а число соседних сигналов возрастает пропорционально K; в результате вероятность возникновения ошибки по крайней мере в одном из передаваемых символов возрастает. В данном случае для передачи одного блока используется энергия, равная KV^2T, а число сигналов равно 2^K.

Далее рассмотрим систему связи с 2^K ортогональными сигналами ψ_i, $i = 1, 2, \ldots, 2^K$, длительности 2^KT; i-й сигнал представляет собой положительный начинающийся в момент i импульс длительности T с энергией KV^2T, который используется для передачи блока из 2^K двоичных символов, представляющего двоичную запись его номера i (фиг. 7.33, *б*). Такие сигналы ис-

пользуются, например, в системах с время-импульсной модуляцией (ВИМ). Один из описанных выше сигналов при $K = 3$ показан на фиг. 7.33, *б*. Как видно из фиг. 7.33, *в*, расстояние между соседними сигналами (в данном случае все сигналы являются соседними) равно $\sqrt{2KTV}$. Поскольку здесь с ростом K число соседних сигналов растет как 2^K, а расстояние между сигналами — как $\sqrt{K}$, то при достаточно больших значениях отношения сигнал/шум ошибку между блоками можно сделать сколь



Фиг. 7.33. Расстояние между блоками.

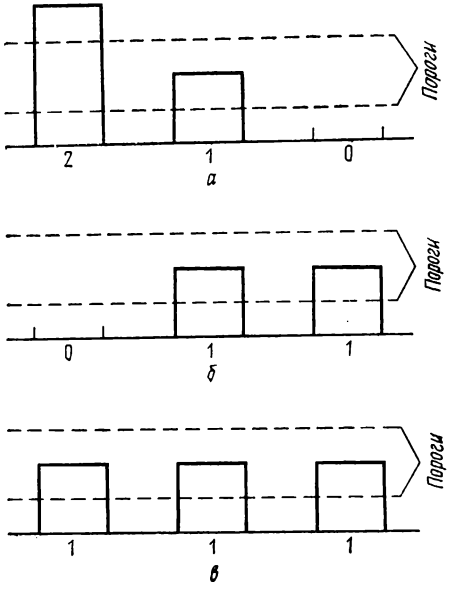
угодно малой, выбрав достаточно большое K. Таким образом, при наличии ограничений лишь на среднюю мощность методы передачи, когда при приеме решение принимается о переданном блоке в целом, позволяют достичь лучших результатов, чем методы передачи, при использовании которых двоичные символы передаются независимо и при приеме решение о каждом из них принимается независимо.

Как показывают приведенные выше примеры, методы принятия решения о символах существенно влияют на вероятность ошибки в символе и посредством этого они связаны с методами исправления ошибок. Многообразие методов принятия решений приводит к многообразию методов исправления ошибок в теории кодирования. Однако в теории кодирования, как правило,

предполагается, что метод принятия решения является жестким и лишь часть получающихся результатов обобщается на системы с исправлением ошибок и стираний.

7.4.3. Расстояние Хэмминга

Одной из условностей, принятых в теории кодирования, является понятие расстояния. Широко используемое в теории кодирования расстояние Хэмминга между векторами $(a_1, a_2, \ldots, a_n)$ и $(b_1, b_2, \ldots, b_n)$ определяется как число пар несовпа-



Фиг. 7.34. Троичные импульсы.

дающих компонент этих векторов. Например, расстояние Хэмминга между векторами (0, 0, 0) и (0, 1, 1) длины 3 с компонентами 0 и 1 равно двум. В то же время равно двум расстояние Хэмминга между троичными векторами (2, 1, 0) и (0, 1, 1) с компонентами 0, 1 и 2. Кроме того, равно двум расстояние Хэмминга между векторами (2, 1, 0) и (1, 1, 1).

Расстояние Хэмминга между двумя векторами определяется лишь числом пар несовпадающих компонент векторов и совершенно не учитывает, насколько сильно различаются несовпадающие компоненты. Слабые стороны такого определения расстояния становятся очевидными, если рассмотреть передачу по каналу с аддитивным гауссовским шумом компонент 0, 1 и 2 тро-

ичных векторов с помощью импульсов фиксированной длительности с амплитудами соответственно 0, 1 и 2 В, как показано на фиг. 7.34, *а*. Предположим, например, что по каналу передается последовательность импульсов, соответствующая вектору (2, 1, 0). В результате воздействия шумов эта последовательность может быть воспринята приемником и как последовательность, соответствующая вектору (0, 1, 1), и как последовательность, соответствующая вектору (1, 1, 1). Однако поскольку шум в канале является аддитивным и гауссовским, то, очевидно, что вероятность принять символ 2 как 0 меньше, чем вероятность принять символ 1 как 0. В то же время расстояние Хэмминга между вектором (2, 1, 0) и векторами (0, 1, 1), (1, 1, 1) одно и то же.

Однако это неестественное с первого взгляда понятие расстояния Хэмминга позволяет упростить модель системы связи, а следовательно, и теоретический анализ последней. Расстоянием, которое определяется не только числом пар несовпадающих компонент, но и учитывает величину различия последних, является расстоянием Ли. Однако сразу следует заметить, что часть теории кодирования, основанная на расстоянии Ли, сравнительно невелика. Если же использовать еще более общее понятие расстояния, а именно евклидово расстояние, как показано на фиг. 7.33, *в*, то построить теорию кодирования оказывается почти невозможным.

7.4.4. Положительные стороны теории кодирования

Таким образом, в теории кодирования достаточно много условностей, но, с другой стороны, она имеет и ряд существенных достоинств. Одно из них заключается в том, что при увеличении длины кода или числа информационных символов, несмотря на то, что число кодовых слов в коде, а следовательно, и число внутренних состояний кодера и декодера растут экспоненциально, для их обработки, как правило, требуется линейно растущее с длиной кода число операций (шагов). Заметим, что в теории конечных автоматов, как правило, экспоненциальное возрастание числа внутренних состояний сопровождается таким же экспоненциальным ростом числа операций (шагов, циклов), необходимых для обработки.

Таким образом, из вышеизложенного следует, что при применении теории кодирования к тем или иным системам необходимо, во-первых, ясно понимать границы применимости этих результатов, а во-вторых, для достижения высокой эффективности и оптимальности систем умело согласовывать методы принятия решений с используемыми кодами, исправляющими ошибки.

7.5. Применения в системах связи метода повторной передачи

Так как обнаружение и исправление ошибок обычно вводятся с целью повышения достоверности передачи данных за счет некоторого увеличения времени декодирования и стоимости системы, то естественно, что решение вопроса о выборе помехоустойчивых кодов и методов их использования в системе связи должно составлять один из этапов проектирования системы в целом. В существующих системах связи для обнаружения и исправления ошибок широко используются различные методы, основанные на повторной передаче данных. Для реализации этих методов обычно требуется достаточно простое устройство обнаружения ошибок, но при этом функционирование схем, необходимых для организации повторной передачи при обнаружении ошибок, оказывается очень тесно связанным с управлением системой связи в целом, что, конечно, является недостатком этих методов.

7.5.1. Системы с полной обратной связью и системы с дублированием передачи

Существуют различные типы систем связи с повторной передачей данных, такие, например, как системы с полной обратной связью, системы с дублированием передачи, системы с обнаружением ошибок посредством продольных или поперечных проверок на четность, системы с обнаружением ошибок посредством циклических кодов и др. В системах с полной обратной связью подлежащие передаче данные посылаются в канал связи и одновременно вводятся в буферное заполняющее устройство или линию задержки. На приемном конце принятый сигнал также запоминается в буферном запоминающем устройстве и возвращается без изменений по каналу обратной связи на передатчик. На передающем конце полученный от приемника сигнал сравнивается с сигналом, хранящимся в буферном запоминающем устройстве, и если сравниваемые сигналы не совпадают, то обычно выполняется одна из следующих операций: 1) формируется сигнал ошибки и передача данных прекращается; 2) на приемник посылается сигнал, указывающий положение ошибки; 3) исходный сигнал передается вторично. Достоинством этой системы является то, что она позволяет без заметного снижения скорости передачи очень эффективно обнаруживать ошибки. Ее недостатками являются низкая эффективность схем, необходимость согласования времени хранения сообщений на передающем конце с задержкой сигнала в линии связи, невозможность использования в сетях передачи данных с коммутацией пакетов.

В системах с дублированием передачи каждый сигнал передается по каналу два раза. На приемном конце принятые сигналы сравниваются, и если они совпадают, то считается, что при передаче ошибок не было. В противном случае на передатчик посылается запрос на повторную передачу. Такая система также очень хорошо обнаруживает ошибки, но скорость передачи информации уменьшается в два раза.

7.5.2. Обнаружение ошибок

Методы обнаружения ошибок с помощью продольных и поперечных проверок на четность или циклических кодов позволяют хорошо обнаруживать ошибки при сравнительно небольшой избыточности, требующейся для их реализации. При использовании продольных и поперечных проверок на четность подлежащие передаче данные располагаются в виде прямоугольной таблицы, к которой добавляются один проверочный столбец и одна проверочная строка так, чтобы число единиц в каждой строке и каждом столбце такой расширенной таблицы было четным (нечетным). Расширенная таблица передается по каналу связи. При приеме подсчитывается число единиц в каждом столбце и каждой строке принятой таблицы и таким образом обнаруживаются ошибки.

В случае использования циклических кодов для обнаружения ошибок подлежащие передаче данные представляются в виде многочлена. Этот многочлен делится на порождающий многочлен $g(X)$ циклического кода, и полученный в результате этого деления остаток передается по каналу вслед за информационными символами, так что кодовый многочлен, коэффициентами которого являются переданные информационные и проверочные символы, делится на $g(X)$. При приеме принятый многочлен делится на $g(X)$, и если остаток от деления оказывается равным нулю, то считается, что при передаче ошибок не было. Если же полученный остаток оказывается отличным от нуля, то это означает, что при передаче в кодовом многочлене возникли ошибки. В этом случае передача данных либо прекращается, либо на передатчик посылается запрос на повторную передачу искаженных данных. Поскольку при обнаружении ошибок с помощью циклических кодов на передающем и приемном концах достаточно иметь лишь схемы деления многочленов, то этот метод может быть реализован технически сравнительно просто. В настоящее время в системах связи с небольшой скоростью передачи обычно используются продольные проверки на четность. В синхронных системах связи с высокой скоростью передачи, как правило, для обнаружения ошибок применяются циклические коды, а для исправления ошибок — повторная передача.

7.6. Применения в системах связи кодов, исправляющих ошибки

7.6.1. Модуляция и кодирование

Системы с обнаружением ошибок почти всегда обладают очень хорошей способностью обнаруживать ошибки и в настоящее время широко используются на практике. Однако для реализации таких систем необходимо иметь обратный канал связи, регулировать время задержки сообщений в буферном запоминающем устройстве; при реализации некоторых из этих систем значительно снижается скорость передачи. Кроме того, в линиях связи между вычислительными машинами при работе последних в режиме «on line» осуществить повторную передачу в случае обнаружения ошибки крайне сложно, поскольку это связано с необходимостью иметь схему задержки на большое время. В подобных случаях для повышения достоверности передачи целесообразнее использовать коды, исправляющие ошибки.

Модель системы связи, которая обычно берется за основу при разработке подсистем контроля и исправления ошибок, показана на фиг. 7.29. Функции каждого блока, представленного на фиг. 7.29, описаны выше. Главная трудность, возникающая при введении этой модели, связана с обоснованием разделения функций модуляции (в широком смысле) и кодирования. В настоящее время даже теоретически трудно установить, насколько это разделение ухудшает эффективность передачи. Однако на практике такое разделение осуществляется, и поэтому реально существуют два круга проблем:

I. Проблемы модуляции (в широком смысле), связанные с выбором типа модуляции (амплитудной, фазовой и т. д.), передачей реальных непрерывных сигналов, обнаружением сигналов, интерференцией, квантованием, искажениями и т. д.

II. Проблемы кодирования, связанные с обнаружением и исправлением ошибок.

Параметрами, характеризующими эффективность цифровой связи, являются отношение энергии принимаемого сигнала, необходимой для передачи одного бита информации, к мощности шумов E_b/N_0, отношение амплитуды сигнала к амплитуде шумов C/N, отношение W/R, устанавливающее связь между полосой и скоростью передачи, и др. Взаимосвязь между этими параметрами и параметрами, характеризующими поведение канала связи (к последним относится, например, вероятность ошибки), определяет эффективность цифровых систем связи.

Джордан [10] показал, что отношение E_b/N_0 характеризует взаимосвязь модуляции и кодирования. Если обозначить через S мощность принимаемого сигнала в системе связи, показанной

на фиг. 7.20, через N_0 мощность белого шума в прямом канале и через $\mathfrak{R}$ скорость передачи информации в битах в секунду, то отношение необходимой для передачи одного бита информации энергии принимаемого сигнала к мощности шумов можно представить в следующем виде:

$$E_b/N_0 = (S/N_0)(1/\mathfrak{R}). \tag{7.29}$$

Далее, если предположить, что каждые T секунд по каналу передается один из M различных сигналов, то скорость передачи информации в битах на один сигнал будет равна

$$R = \mathfrak{R}T \text{ [бит/сигнал]}.$$

Так как энергия E_r одного принимаемого сигнала равна ST, то

$$(E_b/N_0) = (E_r/N_0)(1/R). \tag{7.30}$$

При передаче данных по телефонным каналам скорость передачи информации R определяется интерференцией в канале. В то же время, как указывалось в гл. 6, в каналах связи, в которых применяется последовательное декодирование, скорость передачи R всегда выбирается меньшей вычислительной скорости $R_{\text{выч}}$ с тем, чтобы среднее число операций при декодировании оставалось постоянным. Как видно из формулы (6.19), вычислительная скорость $R_{\text{выч}}$ определяется только переходными вероятностями канала. Вообще говоря, скорость передачи информации всегда ограничена сверху некоторой постоянной, определяемой системой модуляции (в широком смысле) и свойствами канала. Если эту предельную скорость передачи обозначить через $R_{\text{макс}}$ [бит/сигнал], то для каналов без устройств обнаружения и исправления ошибок будет иметь место следующее равенство:

$$E_b/N_0 = \eta = (E_r/N_0)(1/R_{\text{макс}}). \tag{7.31}$$

(Здесь η — некоторая константа, определяемая системой модуляции (в широком смысле), но не зависящая от кодирующих и декодирующих устройств, используемых для обнаружения и исправления ошибок.)

Если ввести кодирование информации с целью обнаружения или исправления ошибок, то скорость передачи будет меньше, чем $R_{\text{макс}}$. При этом величины

$$\alpha = R/R_{\text{макс}} \tag{7.32}$$

и η можно рассматривать как независимые переменные и получить

$$E_b/N_0 = (E_r/N_0)(1/R_{\text{макс}})(R_{\text{макс}}/R) = \frac{\eta}{\alpha}. \tag{7.33}$$

В процессе проектирования систем связи при заданных требованиях к сложности реализации, вероятности ошибки и другим

характеристикам системы желательно использовать минимально возможное значение η и максимально возможное значение α. При проектировании систем связи отношение E_b/N_0 является главным параметром, определяющим взаимосвязь модуляции и кодирования. Отношение E_b/N_0 является главным параметром при передаче информации с помощью маломощных сигналов, как, например, в системах космической связи. В системах наземной связи вследствие перегрузки каналов связи, а также по другим причинам наиболее жесткие требования предъявляются к пиковым напряжениям; требования к мощности передаваемых сигналов являются менее жесткими. В этих случаях главными параметрами являются E_b/N_0 и отношение C/N. Поэтому при проектировании систем связи в соответствии с их целями и назначением желательно учитывать оба параметра E_b/N_0 и C/N. Параметр W/R, характеризующий полосу, для систем космической связи может быть выбран очень большим, а в большинстве систем наземной связи на полосу накладываются ограничения, часто очень жесткие.

7.6.2. Вероятность ошибки при использовании алгебраических кодов

Найдем вероятность ошибки на блок P_B и эквивалентную вероятность ошибки на символ P'_E, которые понадобятся в дальнейшем [6].

Блоковые коды длины n с k информационными символами, исправляющие ошибки кратности t и менее, будем называть (n, k, t)-кодами. Предположим, что при приеме о каждом переданном символе принимается независимое решение. В этом случае вероятность P_B ошибки в кодовом слове (n, k, t)-кода определяется равенством

$$P_B = \sum_{j=t+1}^{n} P_{E_j} = \frac{n!}{(t+1)!\,(n-t-1)!} P_E^{t+1} (1-P_E)^{n-t-1} \times$$

$$\times \left\{1 + \frac{n-t-2}{t-2} \frac{P_E}{1-P_E} + \frac{(n-t-2)\,(n-t-3)}{(t+2)\,(t+3)} \left(\frac{P_E}{1-P_E}\right)^2 + \ldots\right\} <$$

$$< \frac{n!\,P_E^{t+1}}{(t+1)!\,(n-t-1)!} \left\{1 + \frac{n}{t+2} \frac{P_E}{1-P_E} + \frac{n}{(t+2)^2} \left(\frac{P_E}{1-P_E}\right)^2 + \ldots\right\}, \tag{7.34}$$

где

$$P_{E_j} = \binom{n}{j} P_E^j (1-P_E)^{n-j}, \tag{7.35}$$

а P_E — вероятность ошибки при принятии решения о каждом отдельном символе.

Чтобы строго определить эквивалентную вероятность ошибки на символ, необходимо знать полные статистические свойства ошибок, а не только то, что среди n двоичных символов возникает $t+1$ или более ошибок. Вместе с тем каких-либо алгебраических законов, описывающих поведение ошибок для алгебраических кодов, не существует, и поэтому должны исследоваться и определяться вероятности различных конфигураций ошибок. Однако даже при использовании вычислительных машин эта задача весьма трудоемкая, так как требует очень большого числа операций. В данном разделе будет получена верхняя граница для вероятности ошибки на символ при следующих предположениях.

В случае возникновения $t+1$ или более ошибок в результате ошибочных действий декодер может породить еще t ошибок.

Если среди n символов появилось $t+1$ и более ошибок, то $2t+1$ или более ошибок возникают случайно. Предположение о случайном возникновении $2t+1$ или более ошибок вводится по следующим причинам:

I. Почти для всех алгебраических кодов при таком декодировании не существует каких-либо оснований к тому, чтобы результат ошибочных действий сказывался на каких-либо выделенных символах.

II. В процессе декодирования почти всех алгебраических кодов ошибки в информационных символах и ошибки в проверочных символах не разделяются. Следовательно, если результат ошибочных действий связать с некоторыми определенными символами и вероятности ошибок в символах считать различными, то путем расположения проверочных символов на позициях с высокой вероятностью ошибки можно понизить среднюю вероятность ошибки в информационных символах в случае, когда распределение вероятностей ошибок в символах неизвестно.

Итак, при сделанных выше предположениях $x \geqslant 2t+1$ ошибок в n символах возникают независимо, так что среднее число ошибок в информационных символах равно kx/n. Отсюда следует, что среднее число (математическое ожидание) ошибочных информационных символов равно

$$\sum_{j=t+1}^{n-t} \frac{(j+t)\,k}{n} P_{E_j} + k \sum_{j=n-t+1}^{n} P_{E_j}. \tag{7.36}$$

Следовательно, для эквивалентной вероятности ошибки на символ P'_E (после деления на k и замены $(1-P_E)^{n-t-1}$ на 1) полу-

чаем следующую верхнюю границу:

$$P'_E < \frac{(2t+1)}{n} P_E^{t+1} \binom{n}{t+1} \left[\left\{ 1 + \frac{n}{t+2} \frac{P_E}{1-P_E} + \right.\right.$$
$$\left. + \left(\frac{n}{t+2} \frac{P_E}{1-P_E} \right)^2 + \ldots \right\} +$$
$$\left. + \left\{ \frac{1}{2t+1} \frac{n}{t+2} \frac{P_E}{1-P_E} + \ldots + \frac{n-2t-1}{2t+1} \left(\frac{n}{t+2} \frac{P_E}{1-P_E} \right)^{n-t-1} \right\} \right]. \tag{7.37}$$

Вероятность P_E в реальных системах обычно имеет значения порядка $10^{-3} \div 10^{-10}$. В случае когда $nP_E/(t+2)(1-P_E) \ll 1$, т. е. при $P_E \ll (t+2)/(n+t+2)$, правые части формул (7.35) и (7.37) достаточно точно аппроксимируются их первыми или несколькими первыми членами.

Вероятность возникновения ошибки в отдельном символе при передаче по каналу (входящая в эти формулы) зависит от модуляции и должна определяться для каждой системы отдельно.

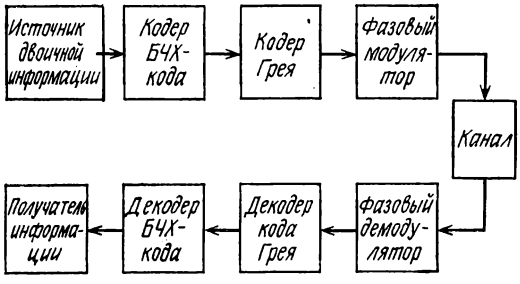
7.6.3. Многоуровневая фазовая модуляция и кодирование

Как указывалось в разд. 7.6.1, установление взаимосвязи между вероятностью ошибки и необходимым для передачи одного двоичного символа отношением энергии принимаемого сигнала к мощности шумов позволяет рассматривать проблемы модуляции и кодирования на единой основе. Ниже мы продемонстрируем еще раз взаимосвязь модуляции и кодирования и покажем, что использование кодирования в системах с многоуровневой фазовой модуляцией позволяет улучшить отношение сигнал/шум при фиксированной вероятности ошибки.

При многоуровневой фазовой модуляции и отсутствии ограничения на выбор отношения сигнал/шум в канале можно повысить $R_{\text{макс}}$ за счет увеличения числа фаз. Однако с увеличением числа фаз увеличивается отношение E_b/N_0, необходимое для достижения заданной вероятности ошибки. Задача состоит в том, чтобы найти точку, в которой возрастание отношения E_b/N_0 за счет увеличения числа фаз компенсируется уменьшением E_b/N_0 за счет введения кодов, исправляющих небольшое число ошибок; решив эту задачу, мы покажем на ее примере эквивалентность систем модуляции (в широком смысле) и систем кодирования при обеспечении помехоустойчивости.

Система связи с многоуровневой фазовой модуляцией и помехоустойчивым кодированием информации показана на фиг. 7.35. Двоичные символы с выхода источника информации сначала поступают на вход блокового кодера, который разбивает их на

блоки и преобразует в кодовые слова некоторого блокового кода, исправляющего ошибки, например БЧХ-кода. Далее последовательность символов с выхода блокового кодера поступает на вход фазового кодера, который преобразует ее в последовательность символов кода Грея, каждый из которых содержит m двоичных информационных символов. Символы кода Грея передаются по каналу посредством импульсов фиксированной длительности с модулированной фазой. При приеме сначала осуществляется синхронное детектирование каждого принятого импульса и далее берется отсчет в максимуме выхода детектора. После декодирования фазовый код преобразуется в двоичный



Фиг. 7.35. Модель системы связи с многоуровневой фазовой модуляцией.

код. С помощью БЧХ-декодера проводится исправление ошибок в двоичных словах, после чего информационные символы направляются получателю информации.

Следуя Кану [7], рассмотрим более детально систему связи с 2^m сигналами вида

$$s(t) = \sqrt{2S}\cos(\omega_0 t + \theta), \quad \theta = \frac{2\pi k}{2^m}, \quad 0 \leqslant k \leqslant 2^m - 1, \tag{7.38}$$

где S — мощность принимаемого сигнала. Предположим, что в канале действует аддитивный белый гауссовский шум

$$n(t) = x(t)\cos\omega_0 t + y(t)\sin\omega_0 t, \tag{7.39}$$

где $x(t)$ и $y(t)$ — низкочастотные случайные компоненты $n(t)$; $\bar{n} = 0$, $\overline{n^2} = N$. В этом случае распределение фазы на выходе

синхронного детектора имеет следующий вид:

$$p(\theta) = \begin{cases} (1/\sqrt{\pi N/S})\, e^{-\theta^2/(N/S)} & |\theta| < \frac{\pi}{2}, \\ \simeq 0, & |\theta| > \frac{\pi}{2}, \end{cases} \tag{7.40}$$

а вероятность ошибки в символе P_s может быть вычислена по формуле

$$P_s = 1 - \int_{-\pi/2^m}^{\pi/2^m} p(\theta)\, d\theta = 1 - 2\,\mathrm{erfc}\left(\frac{\pi}{2^m}\sqrt{\frac{2S}{N}}\right), \tag{7.41}$$

где

$$\mathrm{erfc}\, x = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-u^2/2}\, du. \tag{7.42}$$

Амплитуда сигнала с модулированной фазой при наличии ограничений на ширину полосы не является постоянной и, вообще говоря, канал связи необходимо рассматривать вместе с фильтром. Однако здесь, следуя Беннету и Дэви [12], в целях упрощения при проведении анализа будем считать амплитуду сигнала постоянной.

Фазовый код, задаваемый таблицей Грея, обладает тем свойством, что кодовые слова, соответствующие сигналам с соседними фазами, различаются лишь в одном двоичном символе. При воздействии сильных шумов в результате сильного искажения фазы передаваемого сигнала могут возникать ошибки в двух и большем числе символов кодовых слов кода Грея. Как будет видно ниже, в наиболее важной с практической точки зрения области вероятностей ошибок в отдельном символе, а именно при $P_E \leqslant P_s \simeq 10^{-3} - 10^{-10}$, такими кратными ошибками можно пренебречь.

Положим, например, $m = 4$ и $S/N = 18$ дБ. В этом случае $P_s = 1 - 2 \times 0{,}4860966 = 0{,}278068$, а вероятность ошибки, обусловленная переходом сигнала в соседний, равна $2\,[\mathrm{erfc}\,(6{,}603) - \mathrm{erfc}\,(2{,}201)] = 0{,}0278066$. При этом вероятность ошибки из-за выхода сигнала за пределы соседнего равна $2 \cdot 10^{-7}$, т. е. приблизительно в 10^5 раз меньше вероятности $P_s = 2{,}78 \cdot 10^{-2}$. Аналогично можно убедиться в том, что и во всех остальных случаях в практически важной области вероятностей ошибок в символе $P_E \leqslant P_s \simeq 10^{-3} - 10^{-10}$ с погрешностью, не превышающей 10^{-5}, $P_s = P_E$.

В области вероятностей ошибок в символе, для которых справедливо указанное выше предположение, по аналогии с выводом формул (7.35) и (7.37) в предыдущем разделе для вероятностей ошибок на блок и на символ можно получить сле-

дующие оценки сверху:

$$P_B < \frac{\left[\frac{n}{m}\right]!}{(t+1)!\left(\left[\frac{n}{m}\right]-t-1\right)!} P_E^{t+1}(1-P_E)^{\left[\frac{n}{m}\right]-t-1} \times$$

$$\times\left[1+\frac{\left[\frac{n}{m}\right]}{t+2}\frac{P_E}{1-P_E}+\ldots\right], \quad (7.43)$$

$$P'_E < \frac{(2t+1)}{n} P_E^{t+1}\left(\frac{\left[\frac{n}{m}\right]}{t+1}\right)\times$$

$$\times\left[\left\{1+\frac{\left[\frac{n}{m}\right]}{t+2}\frac{P_E}{1-P_E}+\left(\frac{\left[\frac{n}{m}\right]}{t+2}\frac{P_E}{1-P_E}\right)^2+\ldots\right\}+\right.$$

$$\left.+\left\{\frac{1}{2t+1}\frac{\left[\frac{n}{m}\right]}{t+2}\frac{P_E}{1-P_E}+\frac{2}{2t+1}\left(\frac{\left[\frac{n}{m}\right]}{t+2}\frac{P_E}{1-P_E}\right)^2+\ldots\right\}\right]. \quad (7.44)$$

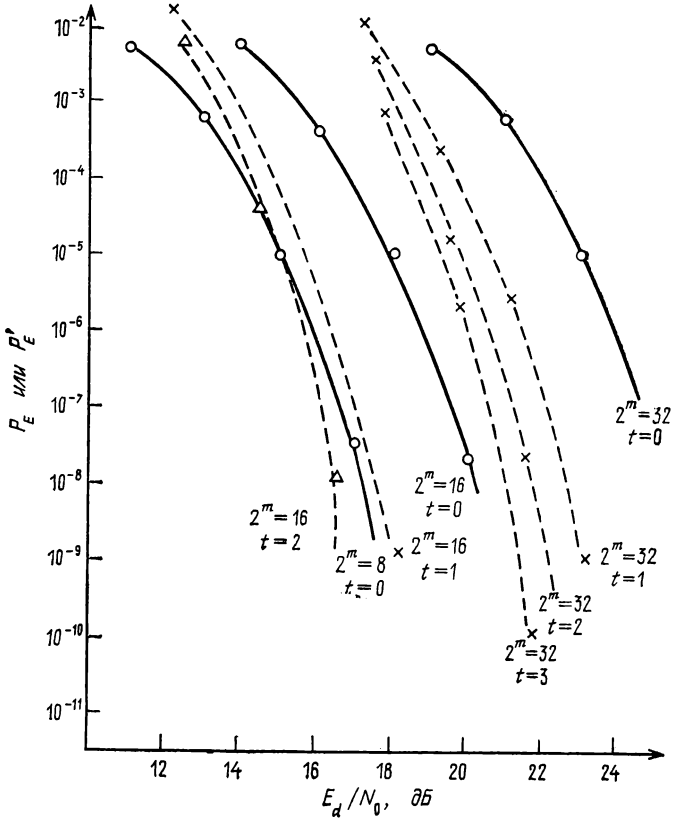
Здесь через $[x]$ обозначено наименьшее целое число, большее или равное x. В случае $P_E \ll (t+2)/\left(\left[\frac{n}{m}\right]+t+2\right)$ правые части последних равенств достаточно точно аппроксимируются первыми членами.

Равенство (7.41) определяет отношение сигнал/шум, необходимое для передачи одного символа. В системах связи с многоуровневой фазовой модуляцией и помехоустойчивым кодированием информации для нахождения отношения сигнал/шум, которое требуется для передачи одного двоичного символа, необходимо внести поправку, обусловленную ростом числа фаз и избыточностью кодов, исправляющих ошибки. При наличии ограничений на ширину полосы поправка вносится путем умножения времени T_B на n/k. Таким образом, если E_b/N_0 вычислять в децибелах, то

$$10\lg(E_b/N_0) = 10\lg(S/N) - 10\lg m + 10\lg(n/k),$$

где $N = N_0 T$, а T — время, необходимое для передачи одного двоичного символа. На фиг. 7.36 приведены результаты вычислений с учетом указанной выше поправки отношения сигнал/шум и вероятности ошибки в случае $k = 100$, $2^m = 8, 16, 32$ ($t = 0$) при использовании БЧХ-кодов с параметрами (127, 120, 1), (127, 114, 2) и (127, 106, 3). Как видно из этих данных, системы без кодирования с $2^m = 8$ и системы с $2^m = 16$, использующие (127, 114, 2)-код, в области практически важных значений P_E почти эквивалентны.

Таким образом, многоуровневая фазовая модуляция использует повышение отношения E_b/N_0 для достижения заданной вероятности ошибки, а методы помехоустойчивого кодирования, основанные на использовании кодов, исправляющих ошибки, допускают снижение отношения E_b/N_0. Использование кодов, исправляющих ошибки, в системах связи с многоуровневой фазо-



Фиг. 7.36. Зависимость вероятности ошибки на символ от E_b/N_0.

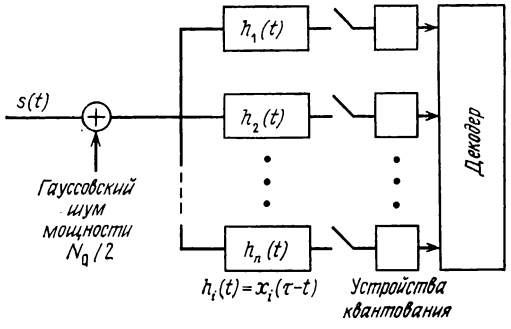
вой модуляцией обеспечивает определенную свободу при проектировании систем благодаря предоставляющейся возможности изменения отношения E_b/N_0 путем изменения числа фаз 2^m.

Рассмотренный выше метод передачи включает преобразование двоичной последовательности, получающейся в результате помехоустойчивого кодирования информационных символов, в сигналы многоуровневой фазовой модуляции. Существует несколько способов реализации такого преобразования, в частно-

сти с помощью использованного выше кода Грея. Другой способ описывается в работе Имаи и др. [8]. Как показывают расчеты, последний способ в некоторых случаях приводит к лучшим результатам, чем код Грея.

7.6.4. Применения кодов в космических и спутниковых системах связи

а) Применение последовательного декодирования. Наиболее ранними работами, в которых начали рассматриваться применения последовательного декодирования в космических системах связи, являются работы Возенкрафта и Кеннеди [9], а также Джордана [10]. Алгоритмам последовательного декодирования



Фиг. 7.37. Декодер максимального правдоподобия для гауссовского канала связи.

и вычислениям их характеристик в этой книге посвящена гл. 6. В данном разделе будут кратко описаны некоторые системы модуляции, используемые в сочетании с последовательным декодированием.

Рассмотрим систему связи, в которой передача последовательности двоичных информационных символов осуществляется блоками по NR символов в каждом. Каждому блоку из NR символов сопоставляется одна из 2^{NR} последовательностей s_l, $l = 1, 2, \ldots, 2^{NR}$, из N сигналов, каждый из которых в свою очередь выбирается кодером из совокупности M ортогональных сигналов x_i, $i = 1, \ldots, M$. Для каждого ортогонального сигнала модулятор формирует сигнал $x_i(t)$, который передается по каналу к приемнику. При приеме, как показано на фиг. 7.37, с помощью фильтров, согласованных с ортогональными сигналами, осуществляется декодирование принятого сигнала по максимуму правдоподобия. Так как выход системы согласованных фильтров представляет собой совокупность M действительных величин,

то его можно представить в виде вектора $\boldsymbol{Y} = (y_1, \ldots, y_M)$. Таким образом, входом рассматриваемого канала являются символы из конечного алфавита, а выходом — M действительных величин. С выходов фильтров эти действительные величины подаются на входы решающего устройства, с помощью которого они вновь преобразуются в дискретные сигналы, поступающие далее на вход декодера. Простейшим решающим устройством является устройство выделения сигнала, имеющего наибольшую апостериорную вероятность. На выходе решающего устройства появляется последовательность $\log_2 M$ двоичных символов.

Как было показано в гл. 6, поведение алгоритмов последовательного декодирования определяется значением $R_{\text{выч}}$. Найдем $R_{\text{выч}}$ для описанной выше системы модуляции. По определению

$$R_{\text{выч}} = \max_{p_k} \left\{ -\log_2 \int_{-\infty}^{\infty} dy \left[\sum_{k=1}^{M} p_k \sqrt{p(\boldsymbol{Y} \mid k)} \right]^2 \right\}, \qquad (7.45)$$

где p_k — вероятность передачи по каналу k-го ортогонального сигнала $(1 \leqslant k \leqslant M)$, $p(\boldsymbol{Y} \mid k)$ — условная плотность распределения вектора $\boldsymbol{Y}$ при условии передачи по каналу k-го ортогонального сигнала. Если предположить, что при фиксированном входе выходы согласованных фильтров являются статистически независимыми, то

$$p(\boldsymbol{Y} \mid k) = p_{s+n}(y_k) \prod_{\substack{l=1 \\ l \neq k}}^{M} p_n(y_l), \qquad (7.46)$$

где $p_{s+n}(y_k)$ — плотность распределения вероятностей на выходе фильтра, соответствующего передаваемому сигналу, а $p_n(y_j)$ — плотность распределения вероятностей на остальных выходах. Подставляя выражение (7.46) в (7.45), после несложных преобразований для $R_{\text{выч}}$ получаем следующую формулу:

$$R_{\text{выч}} = \max_{p_k} \left\{ -\log_2 \left[\sum_{k=1}^{M} p_k^2 + \left(\sum_{\substack{k, l=1 \\ k \neq l}}^{M} p_k p_l \right) \left(\int_{-\infty}^{\infty} \sqrt{p_{s+n}(y)\, p_n(y)}\, dy \right)^2 \right] \right\}. \qquad (7.47)$$

Максимум по p_k в правой части (7.47) достигается при $p_k = 1/M$, $1 \leqslant k \leqslant M$. Подставляя эти значения p_k в формулу

(7.47), находим

$$R_{\text{выч}} = \log_2 \frac{M}{1 + (M-1)\left[\int\limits_{-\infty}^{\infty} \sqrt{p_{s+n}(y)\, p_n(y)}\, dy\right]^2}. \qquad (7.48)$$

При $M \to \infty$ последнее выражение принимает следующий вид:

$$R_{\text{выч}} = -2 \log_2 \left[\int\limits_{-\infty}^{\infty} \sqrt{p_{s+n}(y)\, p_n(y)}\, dy\right]. \qquad (7.49)$$

Далее предположим, что шум в канале является гауссовским и при приеме осуществляется синхронизация фазы. Оптимальный приемник содержит совокупность показанных на фиг. 7.37 согласованных фильтров, с выходов которых берется отсчет в момент окончания передачи каждого ортогонального сигнала. Плотность распределения вероятностей сигнала на выходе фильтра, на который поступают сигнал и шум, и плотность распределения вероятностей сигнала на выходе фильтра, на который поступает только шум, имеют соответственно следующий вид:

$$\begin{aligned} p_{s+n}(x) &= \frac{1}{\sqrt{2\pi}} e^{-(x-a)^2/2}, \\ p_n(x) &= \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \end{aligned} \qquad (7.50)$$

где

$$a = \sqrt{\frac{2E_r}{N_0}}.$$

Подставляя эти выражения в формулу (7.49), для вычислительной скорости в гауссовском канале получаем выражение

$$R_{\text{выч}} = \log_2 \frac{M}{1 + (M-1)\exp(-E_r/2N_0)}. \qquad (7.51)$$

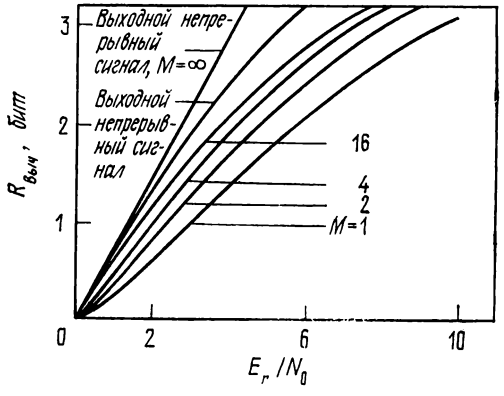
Зависимость $R_{\text{выч}}$ от E_r/N_0 показана на фиг. 7.38. Если с помощью фиг. 7.38 и формулы (7.31) найти $E_b/N_0 = \eta$ и взять достаточно большое значение M, то μ также можно сделать достаточно большим. В пределе при $M \to \infty$ имеем

$$\lim_{M \to \infty} R_{\text{выч}} = \frac{E_r}{2N_0} \log_2 e \text{ [бит/сигнал]}. \qquad (7.52)$$

Это асимптотическое значение $R_{\text{выч}}$ также показано на фиг. 7.38.

Установив таким образом связь между E_b/N_0, вероятностью ошибки и вероятностью переполнения буфера, можно оценить характеристики системы связи с последовательным декодированием.

б) Система модуляции с противоположными сигналами. Одной из систем модуляции, используемых в космических системах связи, является система модуляции с противоположными сигналами, получаемыми путем инверсии фазы. Сравним для этой модуляции характеристики систем с ортогональными кодами и систем, в которых используются алгебраические коды. Для этого рассмотрим систему связи, в которой используются модуляция с инверсией фазы и канал с шириной полосы W_{HZ}, по которому каждые T секунд передается один из $M = 2^n$ возможных сигналов. Предположим, что интервал времени длительностью T секунд, выделенный для передачи одного сигнала, разбивается на n подынтервалов длительностью T/n секунд, каждый из которых



Фиг. 7.38. Вычислительная скорость $R_{\text{выч}}$ при синхронном приеме и $M = 16$.

используется для передачи одного двоичного символа 0 или 1. По каналу для представления символов передается один из следующих двух сигналов длительностью T/n секунд:

$$s_0(t) = \sqrt{2S}\cos(\gamma\pi n/T)\,t, \tag{7.53}$$

$$s_1(t) = -\sqrt{2S}\cos(\gamma\pi n/T)\,t, \tag{7.54}$$

где γ — некоторое неотрицательное целое число. Для реализации этого метода передачи фаза несущей с помощью модулятора изменяется на $\pm 90°$.

Предположим, что при приеме о каждом принимаемом символе выносится независимое решение. Обозначим через p_C вероятность того, что переданный кодовый блок принят правильно, через p_B вероятность того, что переданный кодовый блок принят неправильно, через p_E вероятность ошибки в отдельном символе и через p'_E эквивалентную вероятность ошибки декодирования на символ, которая вычисляется по p_B так, как было описано в

предыдущем разделе. Витерби [21] определяет вероятность ошибки для ортогональных кодов (у которых кодовые блоки попарно ортогональны) и вероятность ошибки в системе без кодирования следующим образом:

А. Вероятность ошибки для ортогональных кодов

$$1 - p_B = p_C = \int_{-\infty}^{\infty} \frac{\exp\left(-v_1^2/2\right)}{\sqrt{2\pi}} \left[\operatorname{erf}\left(v_1 + \sqrt{\frac{2ST}{N_0}}\right)\right]^{M-1} dv_1, \quad (7.55)$$

$$p'_E = \frac{2^{k-1}}{2^k - 1} p_B, \quad (7.56)$$

где

$$\operatorname{erf} x = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-r^2/2}\, dr. \quad (7.57)$$

Б. Вероятность ошибки в системе без кодирования

$$p_E = 1 - \operatorname{erf}\left(\sqrt{\frac{2ST_B}{N_0}}\right). \quad (7.58)$$

При передаче блока из k двоичных символов

$$p_C = 1 - p_B = \left(\operatorname{erf}\sqrt{\frac{2ST_B}{N_0}}\right)^k, \quad (7.59)$$

где

$$T_B = T/k \text{ [с/бит]}. \quad (7.60)$$

Эквивалентная вероятность ошибки на символ при использовании алгебраических кодов определяется равенством (7.37). Однако для нахождения входящей в формулу (7.37) вероятности ошибки в отдельном символе p_E в формулу (7.58) необходимо внести поправку, поскольку из-за расширения полосы шум возрастает в n/k раз. В этом случае

$$p_E = 1 - \operatorname{erf}\sqrt{\frac{2ST_B}{N_0}\frac{k}{n}}, \quad (7.61)$$

а полоса характеризуется параметром

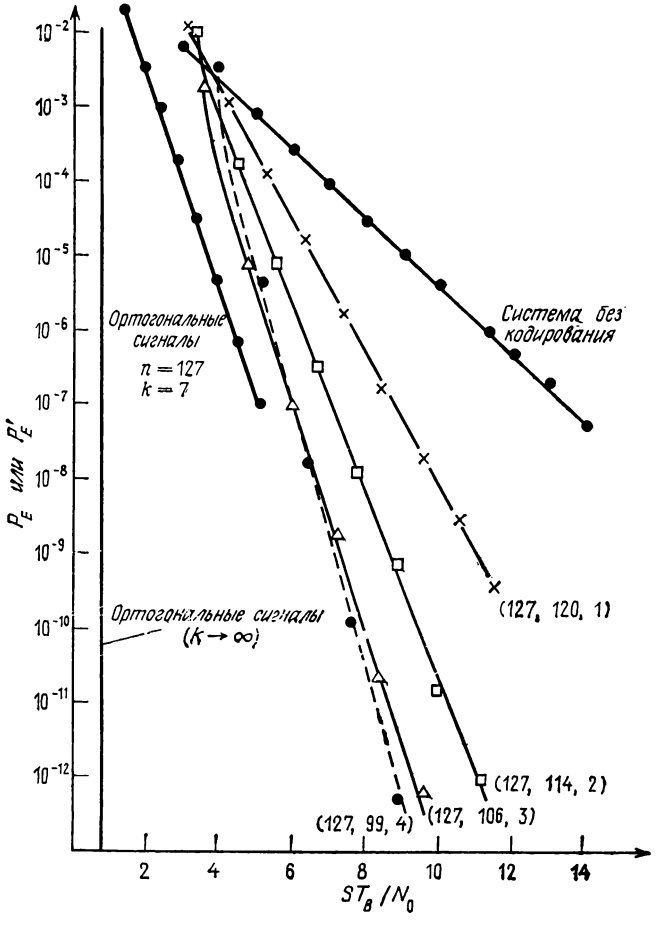
$$W/R = n/2 \log_2 M = n/2k. \quad (7.62)$$

Таким образом,

$$\frac{E_b}{N_0} = \frac{ST_B}{N_0}. \quad (7.63)$$

На фиг. 7.39 показана зависимость вероятности ошибки от ST_B/N_0 для систем связи, в которых используются (127, 120, 1), (127, 114, 2), (127, 106, 3) и (127, 99, 4)-БЧХ-коды, длины $n = 127$ с $k \geqslant 99$, для систем связи без кодирования, для систем связи с ортогональными кодами той же длины $n = 127$ ($k = 7$)

и для систем связи с ортогональными кодами при $k \to \infty$. В случае $n = 127$ и $t = 1$ при $p_E = 10^3$ второй член в формуле (7.35) составляет лишь 4,3%, а при $p_E = 10^{-4}$ — лишь 0,4%. В этой области вероятностей ошибок аппроксимация p_E и p'_E посредством первых членов их разложений (7.35) и (7.37) дает ошибку



Фиг. 7.39. Зависимость вероятности ошибки на символ от ST_B/N_0.

не более 1%. Как видно из фиг. 7.39, при использовании кодов, исправляющих небольшое число ошибок t, заданную вероятность ошибки можно достичь при меньших значениях ST_B/N_0. Выигрыш в отношении ST_B/N_0 обычно выражают в децибелах и называют *выигрышем от кодирования*. Для вероятностей p_E и

p'_E порядка 10^{-6} значения выигрыша от кодирования и необходимой для этого ширины полосы приведены в табл. 7.2. Как видно из этой таблицы, при использовании алгебраических кодов при незначительном увеличении ширины полосы выигрыш от кодирования составляет 2—3 дБ.

Таблица 7.2

Выигрыш от кодирования и необходимая ширина полосы

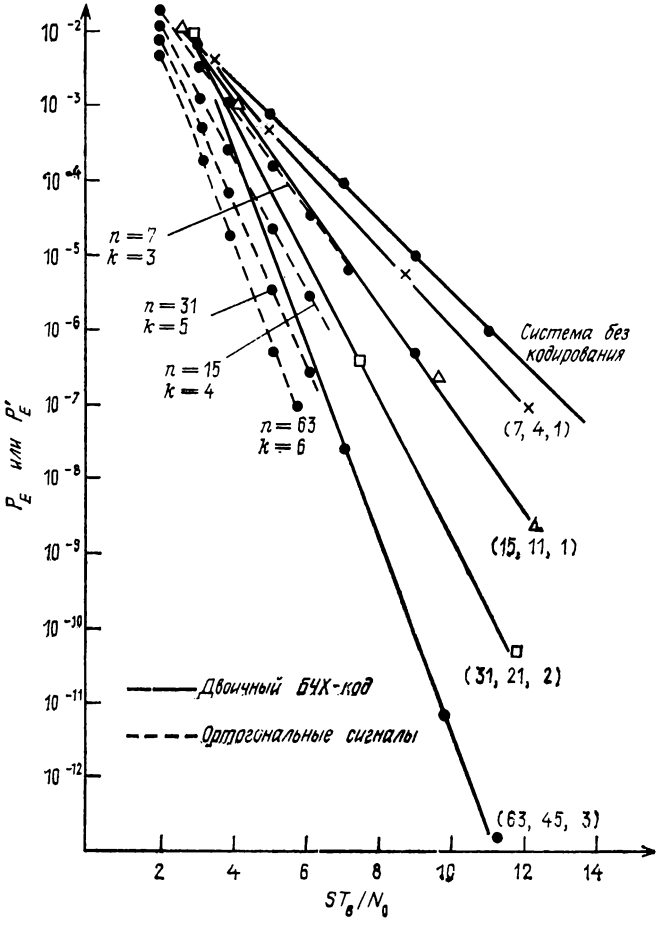
	Без кодирования	$t=1$	$t=2$	$t=3$	$t=4$	Ортогональные коды
Выигрыш от кодирования, дБ	—	1,67	2,53	3,28	3,14	4,13 ($k=7$)
W/R	0,5	0,53	0,55	0,6	0,64	9,0 ($k=7$)

Зависимость вероятности ошибки от отношения ST_B/N_0 при использовании БЧХ-кодов показана также на фиг. 7.40 и 7.41.

Приведенные графики хорошо иллюстрируют взаимосвязь модуляции и кодирования. Для систем без кодирования графики вероятности ошибки были построены в предположении, что при передаче отсутствуют интерференция между словами и другие линейные искажения. При этом предположении поведение кривых вероятности ошибки определяется системами модуляции (в узком смысле) и выделения сигналов. При наличии искажений кривые вероятности ошибки смещаются вверх по отношению к кривой вероятности ошибки для системы без кодирования. Различные методы подавления искажений применяются для того, чтобы приблизить, насколько это возможно, реальную кривую вероятности ошибки к идеальной кривой системы без кодирования. В то же время введение помехоустойчивого кодирования позволяет достичь вероятностей ошибок, лежащих ниже кривой вероятности ошибки системы без кодирования. Сложность аппаратурной реализации кодов с алгебраическим декодированием обычно возрастает линейно или как небольшая степень длины кода n.

Приведенные на фиг. 7.39 графики вероятности ошибки для $t=3$ и $t=4$ в важной с практической точки зрения области значений отношения сигнал/шум почти совпадают, но обычно бывает задан нижний предел числа информационных символов, и интерес представляет поведение вероятности ошибки при увеличении числа проверочных символов. Рассмотрим возможность использования в системах космической связи последовательностей максимальной длины, занимающих в теории кодирования особое положение, а также очень похожих на них равномерных сверточных кодов, введенных в гл. 5.

Совокупность последовательностей максимальной длины является линейным $(2^k - 1,\ k,\ 2^{k-2} - 1)$-кодом, имеющим максимально возможное для своей длины минимальное расстояние. Поэтому эти коды так же, как и ортогональные коды, дают пред-



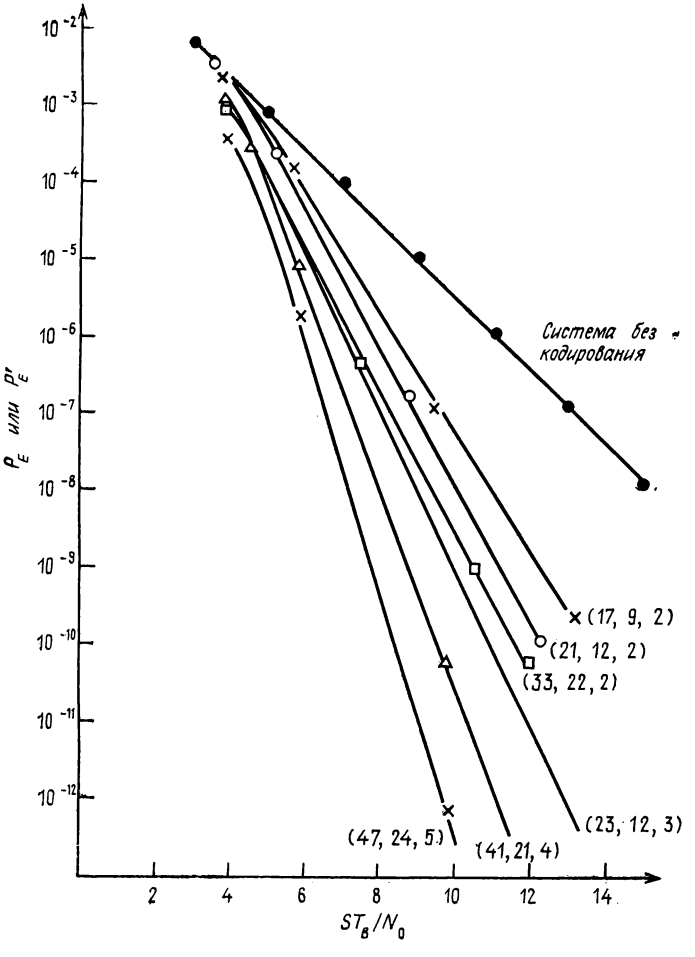
Фиг. 7.40. Зависимость вероятности ошибки на символ от ST_B/N_0.

ставление о предельных возможностях алгебраических кодов с малой скоростью передачи.

Как известно, при $k \to \infty$ отношение ST_B/N_0 для ортогональных кодов стремится к $\ln 2$. Если при приеме решение о каждом переданном символе принимается независимо, то, как следует

из формулы (7.61), вероятность

$$p_E = 1 - \operatorname{erf} \sqrt{\frac{2ST_B}{N_0} \frac{k}{2^k - 1}} \tag{7.64}$$



Фиг. 7.41. Зависимость вероятности ошибки на символ от ST_B/N_0.

стремится к $1/2$. В этом случае, согласно закону больших чисел, с вероятностью, близкой к 1, число ошибок в блоке длины n равно $n/2 = (2^k - 1)/2 \simeq 2^{k-1}$, однако код, состоящий из последовательностей максимальной длины, исправляет только $2^{k-2} - 1$ ошибок. Следовательно, при $k \to \infty$ вероятность ошибки на блок

p_B стремится к 1. Для того чтобы достичь вероятности ошибки $p_B < 1$, необходимо, чтобы выполнялось условие $p_E \leqslant (2^{k-2} - 1)/(2^k - 1) \simeq 1/4$. Это условие может выполняться лишь в том случае, если

$$\operatorname{erf}\left(\sqrt{\frac{2ST_B}{N_0}\frac{k}{2^k-1}}\right) \geqslant \frac{3}{4}, \tag{7.65}$$

т. е. если отношение ST_B/N_0 удовлетворяет следующему неравенству:

$$\frac{ST_B}{N_0} \geqslant 0{,}456\frac{2^k-1}{2k}. \tag{7.66}$$

Последнее неравенство показывает, что при $k \to \infty$ отношение ST_B/N_0 также должно стремиться к бесконечности.

Анализ равномерных сверточных кодов из-за характерного для них явления распространения ошибок оказывается более сложным. Однако поскольку эти коды являются $(k2^{k-1}, k, ((k+1)2^{k-2} - 1)/2)$-кодами, то для достижения вероятности ошибки $p_B < 1$ необходимо, чтобы выполнялось следующее неравенство: $p_E < (k+1)2^{k-2}/k2^k \simeq 1/4$.

Далее для сравнения рассмотрим БЧХ-коды с малой скоростью передачи. Примитивные БЧХ-коды имеют длину $n = 2^m - 1$ и, как следует из границы для БЧХ-кодов, при наличии mt проверочных символов они могут исправлять t и менее ошибок. При $m = k$ примитивные БЧХ-коды являются $(2^k - 1, k, (\lfloor (2^k - 1)/k \rfloor) - 1)$-кодами. Для того чтобы с помощью этих кодов можно было достичь вероятности ошибки $p_B < 1$, необходимо, чтобы выполнялось неравенство $p_E < ((2^k - 1/k) - 1)/2^k - 1 \simeq 1/k$. Это неравенство в свою очередь может выполняться лишь в случае, если

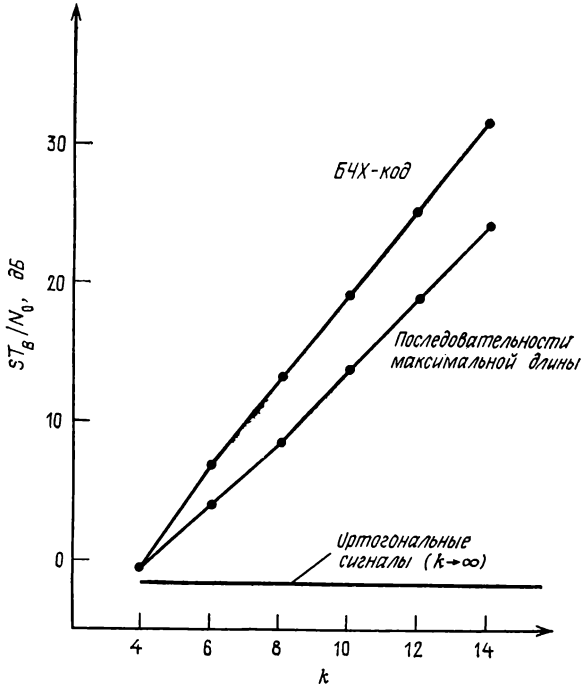
$$\operatorname{erf}\left(\sqrt{\frac{2ST_B}{N_0}\frac{k}{2^k-1}}\right) \geqslant 1 - \frac{1}{k}. \tag{7.67}$$

Однако, как легко видеть, при $k \to \infty$ левая часть (7.67) стремится к $1/2$, т. е. неравенство (7.67) не выполняется. Таким образом, и для этих кодов в пределе также необходимы бесконечно большие значения ST_B/N_0.

Неравенства (7.65) и (7.67) при больших значениях k имеют хорошие приближенные решения; поведение этих решений при малых k, а также при $k \to \infty$ показано на фиг. 7.42. Как видно из фиг. 7.42, во всех рассмотренных выше случаях отношение $ST_B/N_0 \to \infty$ при $k \to \infty$. Это связано с тем, что выражение $k/(2^k - 1)$, входящее в аргумент функции erf в формулах (7.65)

и (7.67), стремится к нулю при $k \to \infty$, что в свою очередь объясняется тем, что при приеме решение о каждом переданном символе принимается независимо.

Применения кодов, исправляющих ошибки, в системах космической связи достаточно многообразны, но что касается систем, рассмотренных выше, то следует заметить следующее. Посимвольное принятие решений и использование алгебраических кодов дают выигрыш 2—3 дБ. Для того чтобы достичь большего



Фиг. 7.42. Отношение ST_B/N_0, при котором $P_B < 1$.

выигрыша, необходимо использовать ортогональные сигналы или последовательное декодирование. Однако при использовании ортогональных сигналов сложность устройств растет экспоненциально в зависимости от длины блока, тогда как при использовании последовательного декодирования теоретически сложность устройств прямо пропорциональна длине кодового ограничения. Поэтому в реальных космических системах связи наиболее целесообразным представляется использование алгебраических кодов или кодов, допускающих последовательное декодирование, в сочетании со сравнительно небольшим числом ортогональных сигналов.

7.6.5. Основные понятия о проектировании систем связи с помехоустойчивым кодированием

Используя результаты, полученные выше в этой и предыдущих главах, остановимся на некоторых основных моментах проектирования систем связи с помехоустойчивым кодированием информации. Сначала рассмотрим системы с исправлением случайных ошибок. Подставляя $R_{\text{макс}}/R = n/k$ в формулу (7.33) и выражая отношение сигнал/шум в децибелах, получаем

$$10 \lg (E_b/N_0) = 10 \lg (E_r/N_0) - 10 \lg R_{\text{макс}} + 10 \lg n/k. \quad (7.68)$$

Рассмотрим по отдельности каждое слагаемое в правой части этого неравенства. Первое слагаемое, как указывалось в разд. 7.6.3 и 7.6.4, связано с системой модуляции, выбором сигналов канала, обнаружением сигналов и т. д. и оказывает влияние на отношение мощности несущей к мощности шумов C/N. Следовательно, если на величину C/N накладываются жесткие ограничения, например, обусловленные высокой загруженностью каналов, то значения C/N оказываются небольшими. Второе слагаемое отражает влияние скорости передачи при фиксированном отношении E_b/N_0 и зависит от выбора сигналов, искажений при передаче, интерференции между кодовыми словами и системы модуляции (в широком смысле). Например, путем компенсации канала можно устранить интерференцию между кодовыми словами и, таким образом, повысить $R_{\text{макс}}$. Третье слагаемое отражает влияние кодирования и показывает, что путем увеличения времени или ширины полосы, используемых для передачи, можно понизить вероятность ошибки.

В тех случаях, когда мощность источника питания передатчика мала, как это имеет место, например, в космической связи, очень важно достичь заданной верности передачи при малых отношениях мощности сигнала (приходящейся на 1 бит передаваемой информации) к мощности шума в канале. Чтобы можно было достичь значительного снижения вероятности ошибки при возможно малых отношениях сигнал/шум, должны быть полностью использованы все возможности, представляемые указанными выше тремя слагаемыми. Как следует из разд. 7.6.3 и 7.6.4, отношение E_b/N_0 можно уменьшить путем использования кодов, исправляющих небольшое число ошибок t. Однако следует заметить, что при больших t отношение E_b/N_0 может увеличиться из-за необходимости дополнительной значительной мощности для передачи избыточных символов или из-за увеличения мощности шума при увеличении ширины полосы. Если решение о каждом переданном символе принимается независимо, то коды с алгебраическим декодированием нельзя заменить на ортогональные сигналы. При использовании ортогональных сигналов

с корреляционным приемом отношение E_b/N_0 при $k \to \infty$ стремится к ln 2, однако при этом с ростом k экспоненциально увеличивается сложность устройств, что, конечно, не приемлемо с практической точки зрения. В реальных космических системах связи наиболее целесообразным представляется использование небольшого числа ортогональных сигналов в сочетании с алгебраическими кодами или последовательным декодированием.

В наземных системах связи (а также в системах космической связи некоторых типов) жесткие мощностные ограничения накладываются сравнительно редко. В этих системах часто имеются жесткие ограничения на отношение C/N, обусловленные перегрузкой диапазонов. Наиболее важной характеристикой систем связи в этих условиях является то, насколько мал первый член в правой части равенства (7.68); малость всей суммы в этом равенстве оказывается менее важным фактором. В подобных случаях для достижения заданной вероятности ошибки целесообразно уменьшить насколько возможно первое слагаемое в правой части (7.68), понизив скорость передачи путем введения соответствующих кодов, исправляющих ошибки, и увеличив третье слагаемое настолько, насколько это позволяет ширина полосы. Следовательно, в этих случаях можно использовать код с небольшим числом исправляемых ошибок t, минимизирующий первый член равенства (7.68), но, быть может, несколько увеличивающий отношение E_b/N_0. Даже при использовании кодов со скоростью $R = {}^1/_2$, когда ширина полосы увеличивается, а скорость передачи уменьшается в два раза, отношение $ST_в/N_0$ увеличивается приблизительно на 3 дБ и существуют достаточно широкие возможности для использования кодов, исправляющих ошибки. Так, например, в приведенном ниже примере 7.6.7 выбор кода, который не приводит к ухудшению отношения E_b/N_0 по сравнению со случаем отсутствия кодирования, позволил улучшить отношение C/N.

Таким образом, важным этапом проектирования оптимальных систем связи является выбор значений трех слагаемых в правой части равенства (7.68) в соответствии с назначением системы связи, требованиями к качеству передачи и, следовательно, к стоимости системы. Введение обнаружения и исправления ошибок, что учитывается третьим членом в правой части (7.68), обеспечивает широкие возможности при проектировании систем связи.

Далее рассмотрим случай, когда шум является импульсным. Импульсный шум обычно имеет амплитуду, значительно превышающую уровень сигнала в канале. Поэтому понизить вероятность ошибки путем повышения мощности сигнала в канале часто не удается. В таких случаях единственным методом борьбы

с импульсным шумом является обнаружение ошибок или использование кодов, исправляющих пачки ошибок.

При введении кодов, исправляющих пачки ошибок, обычно сначала берется модель Гилберта, хорошо описывающая шум в канале, далее выбирается соответствующий код и в завершение оценивается получающаяся вероятность ошибки. Однако импульсные шумы чрезвычайно разнообразны и в некоторых случаях плохо описываются с помощью модели Гилберта. В этом случае одним из способов определения длины кода и длины пачек ошибок, которые должны исправляться для того, чтобы можно было достичь заданной вероятности ошибки, является измерение шума, действующего в канале. Коды, удовлетворяющие определенным таким образом условиям и имеющие минимальную избыточность $m = 2b$, являются оптимальными с точки зрения повышения скорости передачи.

Если длина кода n, найденная по результатам измерений шума в канале, оказывается настолько большой, что реализация соответствующего кода будет очень дорогой, то можно понизить длину n, несколько уменьшив при этом скорость передачи R. Пусть n и b — найденные по результатам измерения шума в канале соответственно длина кода и длина пачек ошибок, которые должны исправляться. Тогда соответствующий код с минимальной избыточностью имеет скорость передачи

$$R = 1 - \frac{2b}{n}. \tag{7.69}$$

Если длину кода уменьшить до величины $n'(<n)$, то скорость передачи кода длины n' с минимальной избыточностью будет равна $R' = 1 - 2b/n'$, и, следовательно, это уменьшение длины кода достигается за счет уменьшения скорости передачи на величину

$$R - R' = 2b\left(\frac{1}{n'} - \frac{1}{n}\right). \tag{7.70}$$

В заключение заметим, что проектирование систем связи, в которых должны использоваться коды, исправляющие пачки ошибок, упрощается при использовании моделирования.

7.6.6. Проблемы, возникающие при проектировании систем связи с помехоустойчивым кодированием информации

В предыдущем разделе кратко были даны основные идеи, используемые при проектировании систем связи в случае применения кодов, исправляющих ошибки. Однако эти идеи в том виде, в каком они были представлены выше, редко применяются при проектировании реальных систем связи. Одной из причин этого является то, что шум в реальных каналах не является чи-

сто гауссовским или чисто импульсным и характеристики канала связи и шум часто изменяются во времени. Например [11], в высокочастотных каналах шум не является ни гауссовским, ни импульсным, а представляет собой смесь гауссовского и импульсного шумов. Известно несколько классов кодов, исправляющих независимые ошибки и пачки ошибок одновременно, но при проектировании систем связи, в которых должны применяться эти коды, пользоваться непосредственно описанными выше принципами нельзя. Кроме того, характеристики затухания в высокочастотных каналах могут изменяться в течение дня. Следовательно, обнаружение или исправление ошибок необходимо проводить в одни периоды времени, тогда как в другие периоды времени необходимости в этом нет. Поэтому при проектировании очень важно использовать системы обнаружения или исправления ошибок, которые можно адаптировать к действующему в канале шуму по результатам его измерения. Приведенные в этой главе некоторые соображения о проектировании следует рассматривать лишь как одни из подходов к решению сложных реальных проблем. При проектировании реальных систем всегда необходимо следить за тем, насколько отвечает реальным условиям получающаяся система.

7.6.7. Пример применения порогового декодирования в спутниковой связи

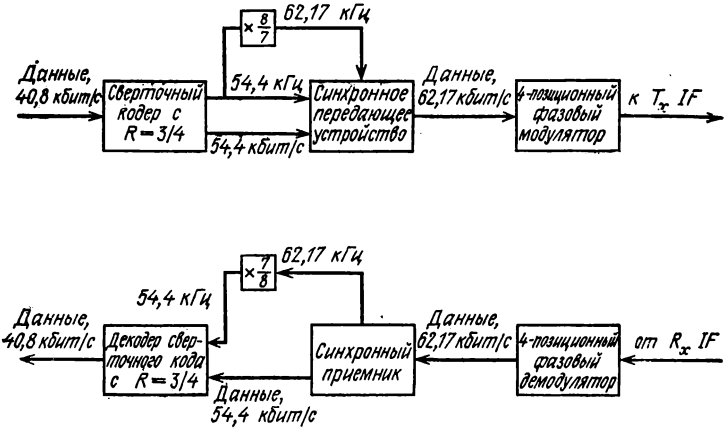
В качестве примера применения теории кодирования рассмотрим систему SPADE, в которой используется пороговое декодирование [13].

Эта система является системой спутниковой связи с несколькими каналами, работающими на разных частотах. При поступлении требования на установление связи каждая станция может использовать любой свободный накал. В этой системе для телефонной связи используется импульсно-кодовая модуляция фазы. Способ связи, допускающий занятие любого свободного канала, позволяет эффективно использовать суммарную пропускную способность каналов спутникового ретранслятора [14].

Для передачи ИКМ-сигнала (который представляет данные, идущие с выхода кодера со скоростью 54 кбит/с) со скоростью на выходе модулятора 64 кбит/с в рассматриваемой системе используется четырехуровневая фазовая модуляция (ФМ) и синхронное детектирование. Скорость передачи информации в битах в секунду равна 32 000.

Эта система предназначена для передачи речи, но после несложной модификации ее можно использовать для передачи данных. На фиг. 7.43 показана схема одного канала этой системы, в котором используется сверточный код со скоростью

$R = {}^3/_4$ и пороговое декодирование. В этом случае на передатчике устанавливается сверточный кодер кода со скоростью $R = {}^3/_4$, а на приемнике — пороговый декодер. Скорости следования символов на входе и выходе кодера составляют соответственно 40,8 и 54,4 кбит/с. В выходную последовательность кодера вводятся синхронизирующие блоки, в результате чего скорость следования символов увеличивается в ${}^8/_7$ раза и на вход модулятора символы поступают уже со скоростью 62,17 кбит/с. Эта скорость лишь несколько меньше 64 кбит/с, так что модулятор и фильтры канала, рассчитанные на скорость 64 кбит/с, используются почти без потерь. В этой системе могут использоваться



Фиг. 7.43. Структура широкополосной системы передачи данных.

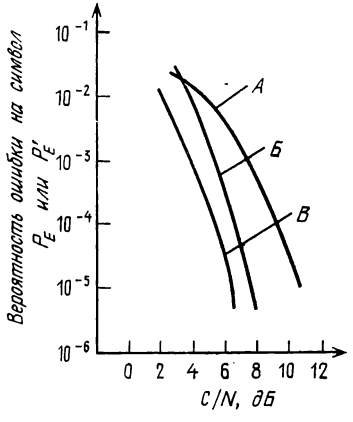
также коды со скоростью $R = {}^1/_2$. В обоих случаях кодеры и декодеры являются достаточно простыми и имеют низкую стоимость. Кодер и декодер кода со скоростью $R = {}^1/_2$ состоят из 17 интегральных схем и стоят ~ 50 долл. Кодер и декодер кода со скоростью $R = {}^3/_4$ состоят из 50 интегральных схем и стоят ~ 250 долл.

Помимо кодов, допускающих пороговое декодирование, применяются коды других типов, например блоковые или сверточные коды в сочетании с последовательным декодированием или декодированием по максимуму правдоподобия (с помощью алгоритма Витерби). В зависимости от типа кода изменяются стоимость и эффективность системы. Естественно, что по мере увеличения сложности кодера и декодера возрастает и их стоимость; так, стоимость последовательного декодера или декодера, реализующего алгоритм Витерби, может в 10—100 раз превышать стоимость порогового декодера.

Для сравнения эффективности различных систем кодирования на фиг. 7.44 приведены результаты измерения вероятности ошибки в зависимости от C/N для трех систем, исследовавшихся с помощью спутника «Интелсат III»:

I. Цифровая передача речи со скоростью 56 кбит/с (ИКМ или Δ-модуляция) или данных, двухуровневая ФМ, синхронное детектирование.

II. Цифровая передача речи со скоростью 28 кбит/с (Δ-модуляция) или данных, двухуровневая ФМ, синхронное детектирование, сверточный код со скоростью $R = {}^1/_2$ и пороговое декодирование.



Фиг. 7.44. Экспериментальные результаты, полученные с помощью спутника «Интелсат III».

А—56 кбит/с, 2-позиционная фазовая модуляция, полоса 80 кГц; *Б*—40,8 кбит/с, $R = {}^3/_4$, пороговое декодирование, 2-уровневая фазовая модуляция, полоса 80 кГц; *В*—28 кбит/с, $R = {}^1/_2$, пороговое декодирование, 2-позиционная фазовая модуляция, полоса 80 кГц.

III. Цифровая передача речи со скоростью 42 кбит/с или данных, двухуровневая ФМ, синхронное детектирование, сверточный код со скоростью $R = {}^3/_4$ и пороговое декодирование.

Как видно из фиг. 7.44, введение кодирования позволяет получить значительный выигрыш в отношении C/N. Указанные выше коды для этого эксперимента были выбраны из-за простоты реализации, низкой стоимости и хорошей эффективности передачи.

7.7. Применение в системах обработки информации

Алгебраические коды, используемые в вычислительных машинах, можно условно разбить на следующие два типа:

1. Коды, исправляющие специфичные для вычислительных машин ошибки, такие, например, как перестановки соседних символов, возникающие при общении человека с машиной, и др.

2. Коды допускающие быстрое декодирование.

К числу кодов типа 1 относятся, в частности, коды Рида — Соломона и некоторые другие хорошо известные коды, которые

используются на практике для исправления различных специфичных для вычислительных машин ошибок [16]. В тех случаях, когда необходимы коды, допускающие быстрое декодирование, циклические коды, занимающие в теории кодирования центральное место, не всегда оказываются наилучшими. Это связано с тем, что алгоритмы декодирования циклических кодов, как правило, являются последовательными, т. е. они сначала исправляют ошибку в некотором одном символе, далее осуществляют циклический сдвиг декодируемого слова, затем исправляют ошибку в следующем символе и т. д. Устройства, реализующие такие алгоритмы, являются очень простыми, но проведение n циклических сдвигов значительно увеличивает время декодирования, что при использовании этих алгоритмов в вычислительной машине снижает эффективность работы последней. В этом разделе будут рассмотрены класс кодов, предназначенных для использования при общении человека с машиной, и класс кодов, допускающих быстрое декодирование.

7.7.1. Применение кодов при общении человека с машиной

В процессе общения человека с машиной иногда не удается точно установить источник возникновения ошибок и принять соответствующие меры к их устранению; в этом случае для повышения надежности обычно используются коды, исправляющие ошибки. Модель процесса общения человека с машиной сложнее, чем модель канала связи.

К числу наиболее типичных ошибок, возникающих в процессе общения человека с машиной, можно отнести следующие: *перестановки* соседних символов, *замена* символов, *вставки* символов, *выпадения* символов и др.

Рассмотрим класс кодов, исправляющих перестановки соседних символов [13]. Перестановку соседних символов всегда можно рассматривать как пачку ошибок длины 2. Так как при перестановке соседних символов значения ошибок в них всегда совпадают, то эти ошибки можно исправить значительно проще, чем обычные пачки ошибок длины 2. Описанные ниже коды являются кодами, которые исправляют ошибку из-за перестановки символов как обычную одиночную ошибку.

Предположим, что передаваемые и принимаемые символы являются элементами поля $GF(q)$. Рассмотрим q-ичный циклический код длины n, задаваемый порождающим многочленом $g(X)$ степени r. Кодовыми словами этого кода являются многочлены степени $n-1$ или менее, которые делятся на $g(X)$. Пусть $g(X)$ является неприводимым многочленом, корнем которого является элемент $\beta = \alpha^{q-1}$, где α — примитивный элемент $GF(q)$.

Так как произвольный вектор-столбец из r символов поля $GF(q)$ всегда можно представить в виде произведения элемента поля $GF(q)$ и одного из столбцов проверочной матрицы рассматриваемого кода, то существуют единственный элемент j поля $GF(q)$ и целое положительное число l, $l < n$, такие, что

$$X - 1 \equiv jX^l \pmod{g(X)}. \tag{7.71}$$

Это сравнение можно переписать в виде

$$\frac{X^{n-l}}{j} \equiv \frac{1}{X-1} \pmod{g(X)}. \tag{7.72}$$

Так как ошибкам из-за перестановки символов соответствует многочлен ошибок вида

$$e_i(X-1)X^l,$$

то синдром определяется равенством

$$s(X) = e_i(X-1)X^l \pmod{g(X)}. \tag{7.73}$$

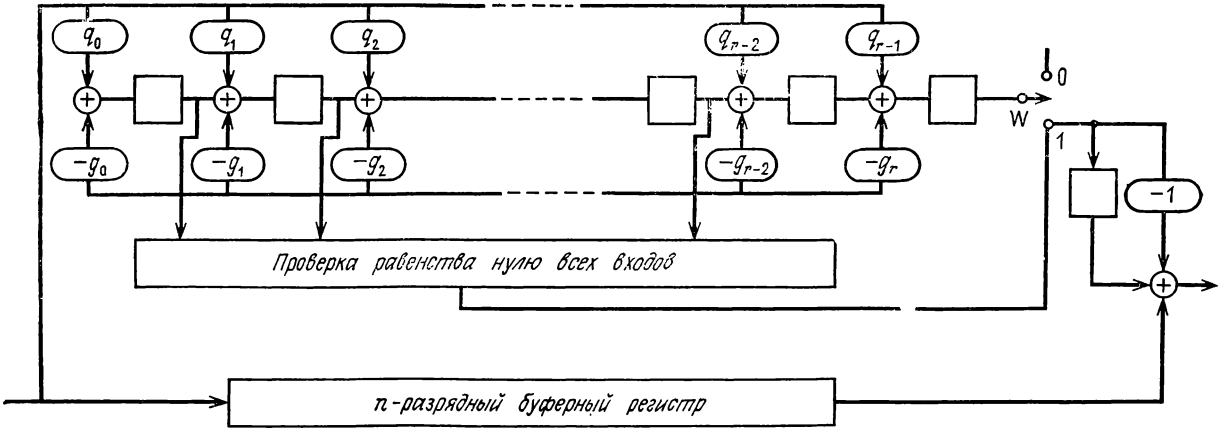
Поскольку $X-1$ и $g(X)$ являются взаимно простыми многочленами, то можно установить взаимно однозначное соответствие между значениями синдрома, соответствующими различным пачкам ошибок, возникающим при перестановке символов, и значениями синдрома, соответствующими простым одиночным ошибкам. Чтобы схему исправления одиночных ошибок можно было использовать для исправления ошибок, возникающих из-за перестановки символов, достаточно воспользоваться следующим соотношением:

$$s'(X) \equiv e_i X^{i+r+1} \equiv \frac{X^{r+1}}{X-1} s(X) \equiv \frac{X^{n-l+r+1}}{j} s(X) \pmod{g(X)}. \tag{7.74}$$

При этом синдром $s'(X)$ можно получить умножением синдрома $s(X)$, вычисленного по принятому кодовому вектору, на многочлен

$$q(X) = q_0 + q_1 X + \ldots + q_{r-1}X^{r-1} \equiv \frac{X^{n-l+r+1}}{j} \pmod{q(X)}. \tag{7.75}$$

Декодер, реализующий описанный выше алгоритм исправления ошибок, возникающих при перестановке символов, показан на фиг. 7.45. Переключатель W находится в положении 0 до тех пор, пока левые $r-1$ разрядов регистра сдвига не будут содержать нулевые символы.



Фиг. 7.45. Декодер, исправляющий ошибки, возникающие при перестановке символов.

7.7.2. Коды на основе ортогональных латинских квадратов

Используя ортогональные латинские квадраты Сяо, Боссен и Чень [14, 18] построили класс кодов для исправления независимых ошибок, ортогонализируемых в один шаг и, следовательно, допускающих пороговое декодирование [14, 18]. Этим кодам посвящен данный раздел. Обычно достичь большой скорости декодирования кодов, исправляющих независимые ошибки, при небольшой избыточности бывает очень трудно. Это связано с тем, что эффективные с точки зрения избыточности коды имеют очень сложные декодирующие устройства, а в результате и низкую скорость декодирования. В подобных случаях часто возникает необходимость повысить скорость декодирования за счет использования более простых алгоритмов декодирования и некоторого увеличения избыточности кода. Пороговые методы декодирования являются как раз такими методами, которые позволяют достичь высокой скорости декодирования. Из особенностей рассматриваемых ниже кодов следует отметить то, что они не являются циклическими и что введение избыточности для повышения корректирующей способности кода осуществляется целыми модулями.

Как указывалось в гл. 5 при описании порогового декодирования, каждый соответствующий информационному символу столбец проверочной матрицы кода, исправляющего t независимых ошибок, должен содержать по крайней мере $2t$ символов 1. Матрицу H, обладающую этим свойством, можно построить на основе латинских квадратов.

Определение 7.1. *Латинским квадратом порядка m называется квадратная матрица размера $m \times m$ с элементами* 0, 1, ..., $m-1$, *каждая строка и каждый столбец которой являются перестановками последовательности* 0, 1, ..., $m-1$. *Два латинских квадрата называются ортогональными, если при наложении друг на друга каждая упорядоченная пара соответствующих элементов этих квадратов появляется ровно один раз.*

Проверочная матрица кода, построенного на основе латинских квадратов, имеет следующий вид:

$$H = \left[\begin{array}{c:c} \begin{matrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{2t} \end{matrix} & I_{2tm} \end{array}\right], \tag{7.76}$$

где I_{2tm} — единичная матрица размера $2tm \times 2tm$,

$$M_1 = \begin{bmatrix} 1 & 1 & \dots & 1 & & & & & 0 \\ & & & & 1 & 1 & \dots & 1 & \\ & & & & & & \cdot & & \\ & & & & & & & \cdot & \\ 0 & & & & & & & 1 & 1 & \dots & 1 \end{bmatrix}_{m \times m^2}, \tag{7.77}$$

$$M_2 = [I_m I_m \dots I_m]_{m \times m^2}, \tag{7.78}$$

а $M_3, \dots, M_{2t}$ — матрицы, которые строятся указанным ниже образом на основе ортогональных латинских квадратов:

$$\begin{aligned} L_1 &= [l_{ij}^1]_{\text{макс}}, \\ L_2 &= [l_{ij}^2]_{\text{макс}}, \\ &\cdot \\ &\cdot \\ &\cdot \\ L_{2t-2} &= [l_{ij}^{2t-2}]_{\text{макс}}, \end{aligned} \tag{7.79}$$

где $l_{ij} \in \{1, 2, \dots, m\}$.

Для каждого из введенных выше латинских квадратов порядка m и каждого элемента латинского квадрата можно построить матрицу инцидентности, которая определяется следующим образом.

Определение 7.2. *Пусть* $L = [l_{ij}]_{\text{макс}}$ — *латинский квадрат, и пусть* μ — *произвольный его элемент. Матрица инцидентности* $Q_\mu = [q_{ij}^\mu]$ *элемента* μ *определяется равенством*

$$q_{ij}^\mu = \begin{cases} 1, & l_{ij} = \mu, \\ 0, & l_{ij} \neq \mu. \end{cases} \tag{7.80}$$

Для каждого латинского квадрата порядка m можно построить m матриц инцидентности $Q_1, Q_2, \dots, Q_m$. Каждой из этих матриц сопоставим вектор

$$V_\mu = [q_{11}^\mu \dots q_{1m}^\mu q_{21}^\mu \dots q_{2m}^\mu \dots q_{m1}^\mu \dots q_{mm}^\mu]. \tag{7.81}$$

Подматрица M_i, $i \geqslant 3$, проверочной матрицы H строится следующим образом;

$$M_i = \begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_m \end{bmatrix}, \tag{7.82}$$

где $V_1, V_2, \ldots, V_m$ — векторы, определяемые формулой (7.81) и соответствующие i-му латинскому квадрату L_i.

Теорема 7.2. *Линейный код с проверочной матрицей H, построенной описанным выше образом по λ_m ($\lambda_m \leqslant m-1$) ортогональным латинским квадратам порядка m, исправляет любые t или менее ошибок, где*

$$t = \frac{\lambda_m}{2} + 1. \tag{7.83}$$

Доказательство. Для доказательства этой теоремы достаточно показать, что проверки, порождаемые матрицей H и контролирующие данный символ, попарно ортогональны относительно последнего. Ортогональность проверок, порождаемых подматрицами M_1 и M_2, очевидна. Далее заметим, что каждый столбец матрицы M_i, $i \geqslant 3$, содержит ровно один ненулевой элемент. Допустим, что некоторая строка матрицы M_i и некоторая строка матрицы M_j пересекаются более чем в одном символе ($i, j = 3, 4, \ldots, 2t$). В этом случае при наложении латинских квадратов L_i и L_j по крайней мере две упорядоченные пары символов будут одинаковыми. Это противоречит определению ортогональности латинских квадратов. Наконец, если допустить, что одна из строк матрицы M_1 (или M_2) совпадает с какой-либо строкой матрицы M_i, $i \neq 1, 2$, то мы придем к тому, что L_i не является латинским квадратом. Вновь получили противоречие. Теорема доказана.

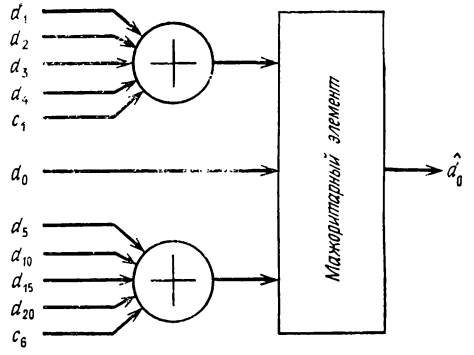
Все коды, которые строятся описанным выше способом, содержат m^2 информационных и $2tm$ проверочных символов.

Пример 7.1. Рассмотрим коды с $k = m^2 = 25$ информационными символами. Используя только матрицы M_1 и M_2, можно построить (35, 25)-код, исправляющий одиночные ошибки. Проверочная матрица H этого кода имеет следующий вид:

$$\left[\begin{array}{ccccccccccccccccccccccccc}
1&1&1&1&1&&&&&&&&&&&&&&&&&&&&1\\
&&&&&1&1&1&1&1&&&&&&&&&&&&&&&&1\\
&&&&&&&&&&1&1&1&1&1&&&&&&&&&&1\\
&&&&&&&&&&&&&&&1&1&1&1&1&&&&&1\\
&&&&&&&&&&&&&&&&&&&&1&1&1&1&1&1\\
1&&&&1&&&&1&&&&1&&&&1&&&&1&&&&1\\
&1&&&&1&&&&1&&&&1&&&&1&&&&1&&&1\\
&&1&&&&1&&&&1&&&&1&&&&1&&&&1&&1\\
&&&1&&&&1&&&&1&&&&1&&&&1&&&&1&1\\
&&&&1&&&&1&&&&1&&&&1&&&&1&&&&1&1
\end{array}\right]. \tag{7.84}$$

Схема декодирования 1-го информационного символа приведена на фиг. 7.46.

Для построения кода, исправляющего двойные ошибки, можно воспользоваться следующими двумя ортогональными латин-



Фиг. 7.46. Декодер (35, 25)-кода исправляющего одиночные ошибки.

скими квадратами:

$$L_1 = \begin{matrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{matrix} \qquad L_2 = \begin{matrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 1 \\ 4 & 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 & 0 \\ 3 & 4 & 0 & 1 & 2 \end{matrix} \tag{7.85}$$

В этом случае код задается проверочной матрицей (7.76), нижние две подматрицы M_3 и M_4 которой имеют следующий вид:

$$M_3 = \begin{bmatrix} 1 & & & & 1 & & 1 & & 1 & & 1 & & \\ & 1 & & 1 & & & & 1 & & 1 & & 1 & \\ & & 1 & & 1 & & 1 & & & & 1 & & 1 \\ & & & 1 & & 1 & & 1 & & 1 & & & 1 \\ & & & & 1 & & 1 & & 1 & & 1 & & \end{bmatrix}, \tag{7.86}$$

$$M_4 = \begin{bmatrix} 1 & & & & 1 & 1 & & & & 1 & 1 & & \\ & 1 & & & & & 1 & 1 & 1 & & & 1 & \\ & & 1 & 1 & & & & & 1 & 1 & & & 1 \\ & & & 1 & 1 & & & & & 1 & 1 & 1 & \\ & & & & 1 & 1 & 1 & & & & 1 & 1 & \end{bmatrix}. \tag{7.87}$$

7.7.3. Коды, исправляющие пачки ошибок и допускающие быстрое декодирование [19]

Несколько модифицировав описанный в гл. 5 метод построения сверточных кодов, исправляющих пачки ошибок, можно построить блоковые коды, исправляющие пачки ошибок, допускающие быстрое декодирование и имеющие параметры $n = b(\sigma + 1) + 2b + \sigma$, $k = b(\sigma + 1)$, $n - k = 2b + \sigma$, $R = b(\sigma + 1)/[b(\sigma + 1) + 2b + \sigma]$. Эти коды не являются циклическими и задаются порождающей матрицей

$$H = \left[\begin{array}{ccc|c|c} I_b & I_b & I_b & I_b & \\ & & & I_b & \\ 0 & 0 & 0 & 0 & I_{2b+\sigma} \\ & & I_b & & \\ & I_b & 00\ldots0 & & \\ I_b & 00\ldots0 & 00\ldots0 & & \end{array}\right], \qquad (7.88)$$

где I_b и $I_{2b+\sigma}$ — единичные матрицы, имеющие соответственно размеры $b \times b$ и $(2b + \sigma) \times (2b + \sigma)$. Декодирование этих кодов точно так же, как и декодирование упомянутых выше сверточных кодов, осуществляется путем выделения в синдроме конфигураций вида (1, 0, ..., 0, 1). Например, код с параметрами $n = 21$, $b = 3$ и $\sigma = 3$ задается проверочной матрицей

$$H = \left[\begin{array}{ccccc} 100 & 100 & 100 & 100 & 100000000 \\ 010 & 010 & 010 & 010 & 010000000 \\ 001 & 001 & 001 & 001 & 001000000 \\ 000 & 000 & 000 & 100 & 000100000 \\ 000 & 000 & 100 & 010 & 000010000 \\ 000 & 100 & 010 & 001 & 000001000 \\ 100 & 010 & 001 & 000 & 000000100 \\ 010 & 001 & 000 & 000 & 000000010 \\ 001 & 000 & 000 & 000 & 000000001 \end{array}\right]. \qquad (7.89)$$

Декодирование этого кода может быть выполнено с помощью следующего алгоритма:

1. Символы $s_1, \ldots, s_9$ синдрома разбиваются на две совокупности (s_1, s_2, s_3) и $(s_4, s_5, \ldots, s_9)$.

2. В случае если $s_3 = 1$, то из конфигураций вида $(s_3, \times, \times, s_6)$, $(s_3, \times, \times, \times\, s_7)$, $(s_3, \times, \times, \times, \times, s_8)$ и

(s_3, $\times$, $\times$, $\times$, $\times$, $\times$, s_9) выбирается та, которая имеет наибольшую длину (знаком $\times$ обозначены символы, значения которых на данном этапе не существенны). С помощью этой конфигурации исправляются ошибки в 12, 9, 6 и 3-м символах. После исправления указанных символов полагаются равными нулю и соответствующие символы синдрома.

3. Из конфигураций вида (s_2, 0, $\times$, s_5), (s_2, 0, $\times$, $\times$, s_6), (s_2, 0, $\times$, $\times$, $\times$, s_7), (s_2, 0, $\times$, $\times$, $\times$, $\times$, s_8) выбирается та, которая имеет наибольшую длину. С помощью этой конфигурации исправляются ошибки в 11, 8, 5 и 2-м символах, после чего также осуществляется коррекция синдрома.

4. Из конфигураций (s_1, 0, 0, s_4), (s_1, 0, 0, $\times$, s_5), (s_1, 0, 0, $\times$, $\times$, s_6) и (s_1, 0, 0, $\times$, $\times$, $\times$, s_7) выбирается та, которая имеет наибольшую длину. С помощью этой конфигурации исправляются ошибки в 10, 7, 4 и 1-м символах, после чего осуществляется коррекция синдрома.

5. Если некоторые символы синдрома отличны от нуля, то осуществляется восстановление.

Предположим, например, что ошибки возникли в 3-м и 5-м символах r_3 и r_5 кодового слова кода, задаваемого порождающей матрицей (7.89). Обозначим через e_i значение ошибки в i-й компоненте кодового слова. В данном случае синдром имеет следующий вид:

$$s = 0e_5e_3000e_50e_3 = 011000101.$$

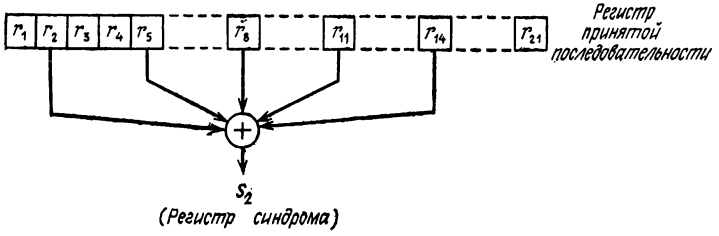
При декодировании принятого кодового слова с помощью описанного выше алгоритма на шаге 2 выделяется конфигурация (s_3, $\times$, $\times$, $\times$, $\times$, $\times$, s_9) и осуществляется исправление ошибки e_3. На шаге 3 выделяется конфигурация (s_2, 0, $\times$, $\times$, $\times$, s_7) и исправляется ошибка e_5.

Рассмотрим декодер, реализующий описанный выше алгоритм декодирования. Этот декодер состоит из схемы вычисления синдрома, схемы исправления ошибок и схемы коррекции синдрома. Схемы кодеров здесь рассматривать не будем, поскольку они аналогичны соответствующим схемам вычисления синдрома, входящим в состав декодеров.

Часть схемы вычисления синдрома, используемая для вычисления символа s_2, приведена на фиг. 7.47. Остальные символы синдрома вычисляются с помощью аналогичных схем.

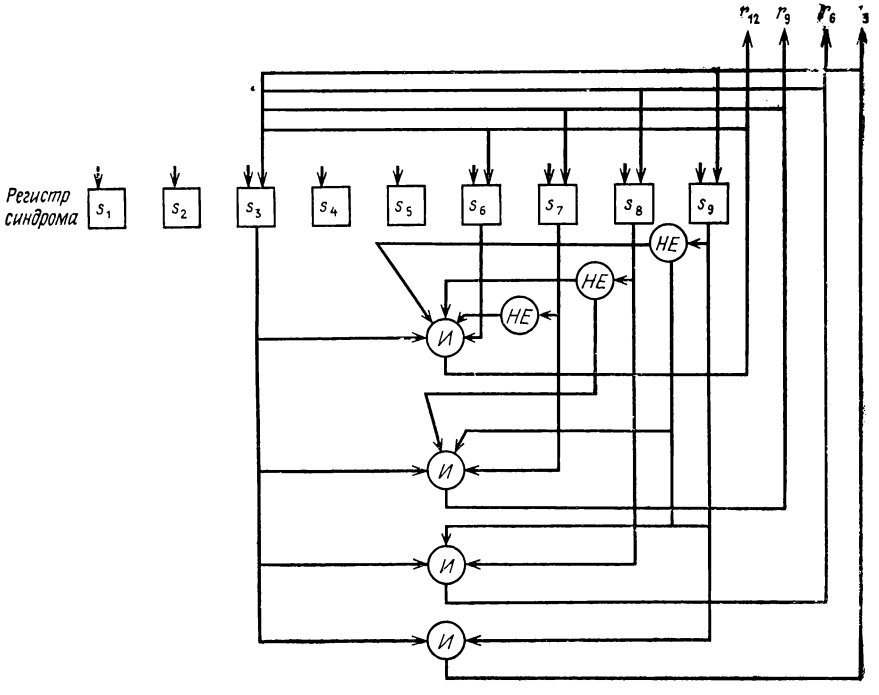
Для реализации шагов 2, 3, и 4 описанного алгоритма используются почти идентичные схемы. На фиг. 7.48 приведена схема, используемая для реализации шага 2. В этой схеме элементы И используются для выделения конфигураций вида (1, $\times$, ..., $\times$, 1), а элементы НЕ — для определения наиболее длинной из них.

Для обеспечения нужной последовательности выполнения операций шагов 2, 3 и 4, а также для реализации шага 5 исполь-



Фиг. 7.47. Часть схемы вычисления синдрома.

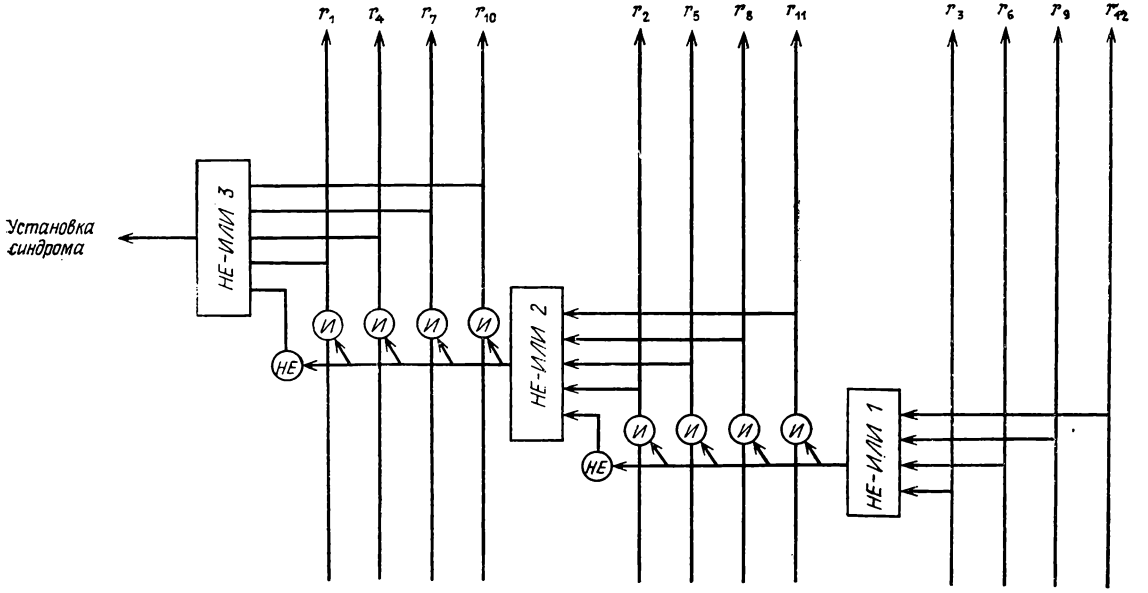
зуется схема, приведенная на фиг. 7.49. Выходы 1 и 2 элементов ИЛИ обеспечивают нужную последовательность выполнения



Фиг. 7.48. Схема, реализующая шаг 2.

операций 2, 3 и 4, а выход 3 элемента ИЛИ используется для реализации шага 5.

В общем случае схема декодера аналогична схеме, описанной выше.



Фиг. 7.49. Схема, устанавливающая последовательность выполнения операций 2, 3, 4 и 5.

Так как рассмотренные здесь коды, исправляющие пачки ошибок, допускают быстрое декодирование, то они могут использоваться в вычислительных машинах. В качестве кодов, исправляющих пачки ошибок и допускающих быстрое декодирование, могут также использоваться коды Файра и их обобщения (алгоритм быстрого декодирования этих кодов предложен Ченем [20]).

Задачи

1. Пусть α — примитивный элемент $GF(2^4)$. Для декодера БЧХ-кодов найти схему вычисления $r(\alpha^7)$.

2. Рассмотрим (псевдо)циклический код длины $n = 14$ с $k = 11$ информационными символами, исправляющий пачки ошибок длины 4 и менее. Исследовать процесс исправления следующей пачки ошибок длины 4: $X^{12} + X^{13} + X^{15}$.

3. Вычислить и сравнить между собой вероятности ошибок на символ при $P_E = 10^{-3}$ и 10^{-5} для кодов со следующими параметрами: а) $n = 23$, $k = 12$, $d = 6$; б) $n = 45$, $k = 23$, $d = 7$.

4. Вычислить и сравнить между собой вероятности ошибок при $P_E = 10^{-3}$ и 10^{-5} для самоортогонального сверточного кода с $R = 3/4$, $J = 6$, $n_A = 248$ и БЧХ-кода с $n = 255$, $k = 191$, $d = 17$.

5. Пусть задан код максимальной длины с $n = 32$, $k = 6$. Найти связь между вероятностью ошибки на символ при алгебраическом декодировании этого кода как циклического кода и отношением E_b/N_0. Установить связь между вероятностью ошибки при корреляционном приеме, когда заданный код рассматривается как биортогональный, и отношением E_b/N_0. Полученные результаты сравнить [21].

6. а) Построить проверочную матрицу кода длины $n = 55$ с $k = 25$, получающегося из ортогональных латинских квадратов и исправляющего три ошибки.

б) Описать аппаратурную реализацию этого кода.

7. а) Найти проверочную матрицу кода с параметрами $n = 33$, $b = 5$, $\sigma = 3$, исправляющего пачки ошибок и допускающего быстрое декодирование.

б) Рассмотреть процесс исправления ошибок в случае возникновения пачки ошибок длины 5.

в) Определить числа ячеек регистров сдвига элементов И, ИЛИ и НЕ, необходимых для построения декодера этого кода.

8. Коды для арифметических устройств

Излагавшаяся до сих пор теория кодирования представляет собой теорию кодов, предназначенных для обнаружения и исправления ошибок в каналах связи, т. е. для повышения надежности связи. Методы повышения надежности вычислений, выполняемых вычислительными машинами и другими устройствами, развивались независимо в двух следующих направлениях.

Первое направление восходит к работам фон Неймана [1] и связано с построением надежных устройств из ненадежных элементов. Результаты развития этого направления были подытожены Виноградом и Коэном [2]. Во всех работах, относящихся к этому направлению, повышение надежности достигается различными методами введения так называемой *структурной избыточности.*

Если задачей первого направления является повышение надежности вычислений путем повышения надежности самого вычислительного устройства, то отличительной особенностью второго направления является то, что само вычислительное устройство не модифицируется, а данные, которые в него вводятся, кодируются так, чтобы на выходе можно было осуществить обнаружение и исправление ошибок, возникших в процессе вычислений. Как показал Элайс [3], такое кодирование эффективно не для всякого вычислительного устройства; однако оно с успехом применяется, например, в устройствах, которые оперируют с целыми числами. В данном случае речь идет об *арифметических кодах*, называемых также AN-кодами, начало исследованию которых было положено работами Дайямонда [4] и Брауна [5]. К числу кодов этого типа относятся также так называемые *коды в остаточных классах* (residue class code) и *линейные вычетные коды* (linear residue code).

В случае AN-кодов сообщения (данные) обычно представляются в виде целых чисел. Кодовыми словами AN-кода являются r-ичные представления чисел-сообщений, умноженных на некоторое фиксированное число (модуль) A, так что кодовое слово суммы и разности двух чисел всегда будет соответственно суммой и разностью кодовых слов, соответствующих исходным числам. Благодаря этому AN-коды могут обнаруживать и исправлять как обычные ошибки, возникающие в каналах связи,

так и ошибки, возникающие в вычислительных устройствах. Поэтому такие коды могут применяться в каналах передачи данных, соединяющих вычислительные машины. В данной книге AN-кодам посвящены эта и следующая главы.

8.1. Основные понятия теории чисел

Для исследования AN-кодов необходимо знать основы теории чисел. Настоящий раздел является вспомогательным и содержит краткое изложение основных понятий и результатов теории чисел, используемых в последующих разделах при рассмотрении AN-кодов. Более детальное изложение этих вопросов можно найти, например, в книгах Такаги [6] и Виноградова [7].

8.1.1. Делимость целых чисел

Пусть $m > 0$ — целое положительное число. Произвольное целое число a единственным образом может быть представлено в виде

$$a = mq + r \quad (0 \leqslant r < m). \tag{8.1}$$

Целое число q называется *частным*, а целое число r — *остатком* от деления a на m или *наименьшим неотрицательным вычетом* по модулю m. Равенство (8.1) иногда записывают также в виде

$$r_m(a) = r, \quad \text{или} \quad a \bmod m = r. \tag{8.2}$$

Очевидно, что в тех случаях, когда m делит a, остаток $r = 0$. В этом случае будем писать $m \mid a$. Для произвольных целых чисел a и b имеют место следующие равенства:

$$\begin{gathered} r_m(a+b) = r_m(r_m(a) + r_m(b)), \quad \text{или} \\ (a+b) \bmod m = (a \bmod m + b \bmod m) \bmod m; \end{gathered} \tag{8.3}$$

кроме того,

$$\begin{gathered} r_m(ab) = r_m(r_m(a)\, r_m(b)), \quad \text{или} \\ (ab) \bmod m = (a \bmod m \cdot b \bmod m) \bmod m. \end{gathered} \tag{8.4}$$

8.1.2. Наименьшее общее кратное и наибольший общий делитель

Общим кратным ненулевых целых чисел $a_1, a_2, \ldots, a_k$ называется любое целое число, которое делится на каждое из этих чисел. Наименьшее положительное из общих кратных называется *наименьшим общим кратным* и обозначается через НОК $(a_1, a_2, \ldots, a_k)$ или $[a_1, a_2, \ldots, a_k]$.

Целое число, которое делит каждое из чисел $a_1, a_2, \ldots, a_k$, называется их *общим делителем*. Наибольший из общих дели-

телей называется *наибольшим общим делителем* и обозначается через НОД $(a_1, a_2, \ldots, a_k)$ или $(a_1, a_2, \ldots, a_k)$.

Если $(a_1, a_2, \ldots, a_k) = 1$, то числа $a_1, a_2, \ldots, a_k$ называются взаимно простыми. Очевидно, что для взаимно простых чисел

$$[a_1, a_2, \ldots a_k] = \prod_{i=1}^{k} a_i = a_1 \cdot a_2 \cdot \ldots \cdot a_k. \tag{8.5}$$

Любое общее кратное двух целых чисел a и b всегда делится на наименьшее общее кратное $[a, b]$. Кроме того, имеет место равенство

$$[a, b] = \frac{ab}{(a, b)}. \tag{8.6}$$

8.1.3. Простые числа и разложение на простые сомножители

Любое целое число $p > 1$ всегда имеет по крайней мере два делителя, а именно 1 и p. Если, кроме этих делителей, число p не имеет других положительных делителей, то оно называется *простым*. Числа, большие 1 и имеющие положительные делители, отличные от 1 и самого себя, называются *составными*.

Произвольное целое число $a > 1$ единственным с точностью до порядка образом разлагается в произведение попарно различных простых чисел p_i (или их степеней $p_i^{\alpha_i}$):

$$a = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k} = \prod_{i=1}^{k} p_i^{\alpha_i}. \tag{8.7}$$

В разд. 9.1 и 9.2 рассматриваются, в частности, разложения такого типа для целых чисел вида $2^n - 1$.

8.1.4. Функция Эйлера

Для любого натурального a *функция Эйлера* $\varphi(a)$ определяется как количество чисел x среди 1, 2, 3, ..., a, взаимно простых с a. Например, $\varphi(1) = 1 (x = 1)$, $\varphi(2) = 1 (x = 1)$, $\varphi(3) = 2 (x = 1, 2)$, $\varphi(4) = 2 (x = 1, 3)$, $\varphi(5) = 4 (x = 1, 2, 3, 4)$, $\varphi(6) = 2 (x = 1, 5)$. Пусть p — простое число. Поскольку все целые положительные числа, меньшие p, взаимно просты с p, то

$$\varphi(p) = p - 1. \tag{8.8}$$

Кроме того, легко проверить, что для любого целого $\alpha > 0$

$$\varphi(p^\alpha) = p^{\alpha-1}(p - 1). \tag{8.9}$$

Если представить произвольное целое a в виде (8.7), то из приведенного ниже свойства мультипликативности функции Эйлера с учетом формул (8.8) и (8.9) получаем

$$\varphi(a) = a\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \ldots \left(1 - \frac{1}{p_k}\right). \qquad (8.10)$$

Упомянутая мультипликативность функции Эйлера означает, что для любых взаимно простых чисел a и b

$$\varphi(ab) = \varphi(a)\,\varphi(b). \qquad (8.11)$$

Еще одно важное свойство $\varphi(a)$ состоит в том, что для произвольного целого числа a

$$\sum_{d \mid a} \varphi(d) = a, \quad \text{или, что то же самое}^{1}), \quad \sum_{0 < d \leqslant a} \varphi\left(\frac{a}{d}\right) = a. \qquad (8.12)$$

Другими словами, сумма значений $\varphi(d)$ по всем d, которые делят a, равна a.

8.1.5. Сравнимость и классы вычетов

Если задано целое число m, называемое *модулем*, то, как можно видеть из формулы (8.1), для любого целого числа a определен остаток $r_m(a) = r$. Если целые числа a и b имеют один и тот же остаток, т. е. если $r_m(a) = r_m(b) = r$, то они называются сравнимыми по модулю m, и это записывается так:

$$a \equiv b \pmod{m}. \qquad (8.13)$$

В тех случаях, когда модуль сравнения понятен из контекста, вместо формулы (8.13) иногда пользуются также упрощенной записью $a \equiv b$. Очевидно, что $a \equiv r$ и $b \equiv r$. В действительности существует бесконечно много чисел, сравнимых с r. Множество $\{a \mid a \equiv r \pmod{m}\}$ всех чисел, сравнимых с r, называется *классом вычетов, содержащим* r. Легко видеть, что существует m попарно непересекающихся классов вычетов в соответствии с m значениями остатка r, $r = 0, 1, 2, \ldots, m-1$. Каждое целое число принадлежит одному из этих m классов вычетов. Если взять по одному представителю каждого из этих классов вычетов, то получится *полная система вычетов по модулю* m. Полная система вычетов является коммутативным кольцом с операциями сложения и умножения по модулю m. Более того, если m является простым числом, то это кольцо является полем. Часто в качестве полной системы вычетов используется совокупность наименьших неотрицательных остатков $0, 1, 2, 3, \ldots, m-1$.

[1]) В этой формуле предполагается, что $\varphi(x) = 0$ для нецелых x. — *Прим. ред.*

Если некоторый класс вычетов по модулю m содержит число, взаимно простое с m, то и все целые числа, входящие в этот класс вычетов, будут взаимно простыми с m. Как легко видеть из определения функции Эйлера, существует $\varphi(m)$ таких классов вычетов. Если у каждого из этих классов вычетов взять по одному представителю, то получится *приведенная система вычетов по модулю m*. Приведенная система вычетов является группой относительно операции умножения по модулю m.

8.1.6. Теоремы Эйлера и Ферма

Теорема Эйлера. *Если* $m > 1$ *и* $(a, m) = 1$,

$$a^{\varphi(m)} \equiv 1 \pmod{m}, \tag{8.14}$$

т. е. $a^{\varphi(m)} - 1$ *делится на* m.

Эта теорема является обобщением следующей теоремы Ферма.

Теорема Ферма. *Если* p *— простое число и* a *не делится на* p, *то*

$$a^{p-1} \equiv 1 \pmod{p}. \tag{8.15}$$

Умножив обе части последнего равенства на a, получим

$$a^p \equiv a \pmod{p}.$$

Последнее равенство справедливо для любого a.

8.1.7. Показатели экспоненты и примитивные корни

Пусть $(a, m) = 1$. Рассмотрим бесконечную последовательность

$$a^1,\ a^2,\ a^3,\ a^4,\ a^5,\ \ldots.$$

Согласно теореме Эйлера, существует целое положительное число e, такое, что $a^e \equiv 1 \pmod{m}$. Минимальное целое положительное число e, удовлетворяющее этому условию, называется *показателем* числа a по модулю m. Иногда e называется также *порядком* a по модулю m. Показатель a по модулю m обозначают следующим образом:

$$e = E(a, m).$$

Если $e = E(a, m)$, то числа $1 = a^0,\ a^1,\ a^2,\ \ldots,\ a^{e-1}$ являются попарно несравнимыми по модулю m. Далее, если $a^k \equiv 1 \pmod{m}$, то e делит k. Очевидно, что e делит $\varphi(m)$. Наконец, если показатель a по модулю m равен e и $(k, e) = d$, то показатель a^k по модулю m равен e/d.

Для примера в табл. 8.1 указаны вычеты (остатки) по модулю $m = 45$, взаимно простые с m (т. е. дана приведенная си-

Таблица 8.1

Приведенная система вычетов по модулю $m = 45$ и показатели этих вычетов

$e = E(a, 45)$	$\{a \mid a^e \equiv 1 \pmod{45}, (a, 45) = 1\}$
1	1
2	19 26 44
3	16 31
4	8 17 28 37
6	4 11 14 29 34
8	—
12	2 7 13 22 23 32 38 43
24	—

стема вычетов) и их показатели. В этом случае $\varphi(45) = 24$ и из таблицы видно, что не существует целых чисел с показателем $\varphi(45)$. Это означает, что никакое число, принадлежащее приведенной системе вычетов по модулю 45, не порождает эту систему. Другими словами, не существует такого целого числа a, что $\{a^1, a^2, a^3, \ldots, a^{\varphi(45)}\}$ является приведенной системой вычетов по модулю 45. Кроме того, не существует и целого числа, которое имело бы показатель 8 по модулю 45. Таким образом, следует заметить, что не каждый делитель числа $\varphi(45)$ является показателем какого-либо целого числа по модулю 45.

Если показатель a по модулю m равен $\varphi(m)$, т. е. если $\varphi(m) = E(a, m)$, то a называется *примитивным элементом* по модулю m. Как показывает приведенный выше пример, примитивные элементы существуют не для всех целых чисел. Известно, что примитивные элементы существуют только для целых чисел m вида

$$m = 2,\ 4,\ p^\alpha,\ 2p^\alpha,$$

где p — простое нечетное число и $\alpha \geqslant 1$. Более того, для каждого из этих чисел существует $\varphi(\varphi(m))$ примитивных элементов. Если g является примитивным элементом по модулю p и $g^{p-1} \not\equiv 0 \pmod{p^2}$, то g является примитивным элементом также и по модулю p^α $(\alpha \geqslant 1)$. Кроме того, если g — один из примитивных элементов по модулю m, то все примитивные элементы по модулю m представляют собой числа $g^k \bmod m$, $(k, \varphi(m)) = 1$.

Например, минимальным положительным примитивным элементом по модулю $p = 13$ является 2. Следовательно,

$$2^{12} \equiv 1 \pmod{13}.$$

Действительно, как видно из табл. 8.2, элементы $2^i \bmod 13$ образуют приведенную систему вычетов по модулю 13.

Таблица 8.2

i	1	2	3	4	5	6	7	8	9	10	11	12
$2^i \bmod 13$	2	4	8	3	6	12	11	9	5	10	7	1

Так как $\varphi(\varphi(13)) = \varphi(12) = 4$, то существуют четыре примитивных корня для 13, а именно $2^1 \equiv 2$, $2^5 \equiv 6$, $2^7 \equiv 11$, $2^{11} \equiv 7$.

Необходимое условие существования в приведенной системе вычетов по модулю m числа a, принадлежащего показателю e, можно получить, используя определенную ниже *λ-функцию Кармайкла*. Эта функция определяется на множестве целых чисел m следующим образом:

1) если $m = 2^n$, $n = 0, 1, 2$, то $\lambda(m) = \varphi(m)$;

2) если $m = 2^n$, $n > 2$, то $\lambda(m) = \varphi(m)/2$;

3) если $m = p^\alpha$, $\alpha > 0$, где p — простое число, то $\lambda(m) = \varphi(p^\alpha)$.

4) если $m = 2^n p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$, $n \geqslant 0$, $\alpha_i > 0$, и p_i — попарно различные нечетные простые числа, то

$$\lambda(m) = \text{НОК}\left(\lambda(2^n),\ \lambda\left(p_1^{\alpha_1}\right),\ \dots,\ \lambda\left(p_k^{\alpha_k}\right)\right).$$

Из определения $\lambda(m)$ и теоремы Эйлера следует, что

$$a^{\lambda(m)} \equiv 1 \pmod m. \tag{8.16}$$

Известно, что целое число $a < m$, взаимно простое с m и имеющее показатель e (другими словами, элемент a приведенной системы вычетов показателя e), существует тогда и только тогда, когда e является делителем $\lambda(m)$. Например, при $m=45$

$$\lambda(m) = \lambda(45) = \text{НОК}(\lambda(3^2),\ \lambda(5)) = \text{НОК}(6,\ 4) = 12,$$

и для каждого делителя числа $\lambda(45) = 12$ (и только для таких делителей) существует целое число, взаимно простое с m и имеющее своим показателем этот делитель. Это согласуется с табл. 8.1.

Если $e = E(a, m)$, $(a, m) = 1$ и $m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$, то

$$E(a, m) = \text{НОК}\left(E\left(a, p_1^{\alpha_1}\right),\ E\left(a, p_2^{\alpha_2}\right),\ \dots,\ E\left(a, p_k^{\alpha_k}\right)\right). \tag{8.17}$$

8.1.8. Мультипликативные группы и их разложение

Приведенная система вычетов по модулю m является коммутативной группой с операцией умножения по модулю m. Обозначим эту группу через $G(m)$, а именно

$$G(m) = \{r \mid 0 < r < m,\ (r, m) = 1\}. \tag{8.18}$$

Если m — нечетное число, то в $G(m)$ входит число 2. Если показатель 2 по модулю m равен δ, т. е. если $2^\delta \equiv 1 \pmod m$, то δ является делителем $\lambda(m)$, а следовательно, и делителем $\varphi(m)$. Множество $\{2^0, 2^1, \ldots, 2^{\delta-1}\}$ степеней числа 2, вычисленных по модулю m, образует подгруппу порядка δ группы $G(m)$. Как хорошо известно, группу $G(m)$ можно разложить на непересекающиеся смежные классы по этой подгруппе. Число таких смежных классов будет в точности равно $\varphi(m)/\delta$.

Рассмотрим, например, разложение группы $G(105)$ ($m = 105$). В этом случае $m = 105 = 3 \cdot 5 \cdot 7$ и $\varphi(105) = 48$, т. е. в $G(105)$ входит 48 целых чисел. Показатель числа 2 по модулю 105 определяется из соотношений

$$E(2, 105) = \text{НОК}(E(2, 3), E(2, 5), E(2, 7)) = \text{НОК}(2, 4, 3) = 12.$$

Разложение группы $G(105)$ на $\varphi(105)/12 = 48/12 = 4$ смежных класса по подгруппе $\{2^i \bmod 105 \mid i = 0, 1, \ldots, 11\}$ порядка 12 задается табл. 8.3.

Таблица 8.3

Разложение G (105) по подгруппе

$\{2^i \bmod 105 \mid i = 0, 1, \ldots, 11\}$

2^i	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
$2^i \bmod 105$	1	2	4	8	16	32	64	23	46	92	75	53
$11 \cdot 2^i \bmod 105$	11	22	44	88	71	37	74	43	86	67	29	58
$13 \cdot 2^i \bmod 105$	13	26	52	104	103	101	97	89	73	41	82	59
$17 \cdot 2^i \bmod 105$	17	34	68	31	62	19	38	76	47	94	83	61

Предположим опять, что m — нечетное число. Если m не является простым, то существуют вычеты по модулю m, не являющиеся взаимно простыми с m. Пусть α — один из этих вычетов. Заметим, что α не является элементом $G(m)$. Однако если $(\alpha, m) = d$, то $(\alpha/d, m/d) = 1$ и, следовательно,

$$\frac{\alpha}{d} \in G\left(\frac{m}{d}\right).$$

Группу $G(m/d)$ точно так же, как и $G(m)$, можно разложить на $\varphi(m/d)/\delta'$ смежных классов по подгруппе $\{2^0, 2^1, \ldots, 2^{\delta'-1}\}$ порядка δ', где δ' — показатель числа 2 по модулю m. Эту процедуру можно выполнить для всех делителей d модуля m/d. При этом исходный вычет α может быть найден путем умножения вычета, принадлежащего $G(m/d)$, на делитель d. Таким образом, для произвольного нечетного m множество вычетов по модулю m (т. е. полная система вычетов) разлагается на смежные классы, не имеющие общих элементов. Отсутствие общих элементов следует из формулы (8.12) и того, что при $d_1 \neq d_2$ множества $d_1G(m/d_1) = \{d_1a \mid a \in G(m/d_1)\}$ и $d_2G(m/d_2)$ не имеют общих элементов. Следует также заметить, что поскольку показатели числа 2 по модулю m/d для разных d различны, то в разных смежных классах содержится неодинаковое число элементов.

Пусть, например, $m = 105$ и $d = 3$. Рассмотрим мультипликативную группу $G(105/3) = G(35)$. В данном случае $\delta' = E(2, 35) = 12$, $\varphi(35) = 24$, и, следовательно, $G(35)$ состоит из 24 элементов. Группа $G(35)$ разлагается на $24/12 = 2$ смежных класса, как показано в табл. 8.4.

Таблица 8.4

Разложение G (35)

2^i	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
$2^i \bmod 35$	1	2	4	8	16	32	29	23	11	22	9	18
$3 \cdot 2^i \bmod 35$	3	6	12	24	13	26	17	34	33	31	27	19

Соответствующие вычеты α по модулю 105, такие, что $(\alpha, 105) = 3$, представлены в табл. 8.5.

Таблица 8.5

Вычеты по модулю 105, соответствующие двум смежным классам группы G (35)

$3 \cdot 2^i \bmod 105$	3	6	12	24	48	96	87	69	33	66	27	54
$9 \cdot 2^i \bmod 105$	9	18	36	72	39	78	51	102	99	93	81	57

Аналогично, рассматривая мультипликативную группу $G(105/21) = G(5)$ при $d = 21$, получаем, что $E(2, 5) = 4$, $\varphi(5) = 4$. Группа $G(5)$ из четырех элементов и соответствующие вычеты по модулю 105 приведены соответственно в табл. 8.6а и 8.6б.

Таблица 8.6а

G (5)

2^i	2^0	2^1	2^2	2^3
$2^i \bmod 5$	1	2	4	3

Таблица 8.6б

Вычеты по модулю 105, соответствующие G (5)

$21 \cdot 2^i \bmod 105$	21	42	84	63

8.1.9. Квадратичные вычеты и символ Лежандра

Пусть a и m — такие целые числа, что $(a, m) = 1$. Если сравнение

$$x^2 \equiv a \pmod{m} \tag{8.19}$$

имеет решение в целых числах, то число a называется *квадратичным вычетом* по модулю m. В противном случае, т. е. если (8.19) не имеет решения в целых числах, число a называется *квадратичным невычетом*.

Для того чтобы целое число a было квадратичным вычетом по модулю простого числа p, необходимо и достаточно, чтобы выполнялось следующее сравнение:

$$a^{(p-1)/2} \equiv 1 \pmod{p}. \tag{8.20}$$

Для всех целых чисел a, которые не делятся на p, определим величину (a/p) следующим образом:

$$\left(\frac{a}{p}\right) = \begin{cases} 1, & \text{если } a \text{ — квадратичный вычет по модулю } p, \\ -1, & \text{если } a \text{ — квадратичный невычет по модулю } p. \end{cases} \tag{8.21}$$

Введенная величина называется символом Лежандра. Имеет место следующее сравнение:

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p} \quad \text{(критерий Эйлера)}, \tag{8.22}$$

включающее как частный случай также сравнение (8.20). Для произвольных нечетных простых чисел p и q справедливы также следующие законы:

1) Первое дополнение к закону взаимности

$$\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}. \tag{8.23}$$

2) Второе дополнение к закону взаимности

$$\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/2}. \tag{8.24}$$

3) Закон взаимности

$$\left(\frac{q}{p}\right)\left(\frac{p}{q}\right) = (-1)^{\{(p-1)/2\}\{q-1)/2\}}. \tag{8.25}$$

В заключение сформулируем еще одно необходимое и достаточное условие для произвольного модуля $m = 2^{\alpha} p_1^{\alpha_1} p_2^{\alpha_2} \ldots$ $\ldots p_k^{\alpha_k}$, где $p_1, p_2, \ldots, p_k$ — нечетные простые числа. Для того чтобы число a, взаимно простое с модулем, было квадратичным вычетом, необходимо и достаточно, чтобы выполнялись следующие условия: 1) $a \equiv 1 \pmod 4$, если $\alpha = 2$, и 2) $a \equiv 1 \pmod 8$ и $(a/p_1) = 1$, $(a/p_2) = 1, \ldots, (a/p_k) = 1$, если $\alpha \geqslant 3$. Кроме того, заметим, что если 2 является квадратичным вычетом по модулю p, то 2 будет также и квадратичным вычетом по модулю p^{α} $(\alpha > 1)$.

8.1.10. Круговые многочлены и их сомножители

Как хорошо известно, алгебраическое уравнение

$$x^d - 1 = 0 \tag{8.26}$$

имеет d различных корней $\alpha_1, \alpha_2, \ldots, \alpha_d$. Эти корни называются *корнями степени d* из 1 и часто представляются в виде комплексных чисел:

$$\alpha_m = \cos\frac{2\pi m}{d} + j\sin\frac{2\pi m}{d} \quad (m = 0, 1, 2, \ldots, d-1). \tag{8.27}$$

Если корень α_m обладает тем свойством, что все элементы α_m^0, $\alpha_m^1, \ldots, \alpha_m^{d-1}$ различны, то он называется *примитивным корнем степени d* из 1. Элемент α_m является примитивным корнем степени d из 1 тогда и только тогда, когда $(m, d) = 1$. Следовательно, всего существует $\varphi(d)$ примитивных корней из 1. Если α_m является примитивным корнем степени d из 1, а кроме того, и корнем некоторого уравнения $x^n - 1$, то число d должно делить n.

Если $\alpha_0, \alpha_1, \ldots, \alpha_k, \ldots, \alpha_{n-1}$ — различные корни n-й степени из 1 [1]), то

$$x^n - 1 = (x - \alpha_0)(x - \alpha_1) \ldots (x - \alpha_k) \ldots (x - \alpha_{n-1}). \quad (8.28)$$

Те из корней α_k, для которых $(k, n) = 1$, являются примитивными корнями n-й степени из 1. Каждый из остальных корней является корнем некоторой степени d из 1, где d — один из делителей числа n $(1 < d < n)$. Пусть $\Phi_d(x)$ — многочлен от x, корнями которого являются только примитивные корни d-й степени из 1, а именно пусть

$$\Phi_d(x) = \prod_{(k,\, d)=1} (x - \alpha_k), \quad (8.29)$$

где $\prod_{(k,\, d)=1}$ — произведение по всем примитивным корням d-й степени из 1 [2]).

Заметим, что $\Phi_d(x)$ имеет степень $\varphi(d)$. Поскольку каждый корень n-й степени из 1 является примитивным корнем степени d из 1 для некоторого делителя d числа n, то

$$x^n - 1 = \prod_{d \mid n} \Phi_d(x), \quad (8.30)$$

где $\prod_{d \mid n}$ — произведение по всем делителям d числа n.

Многочлен $\Phi_d(x)$ называется *круговым многочленом* порядка d. Он обладает рядом интересных свойств, однако здесь для нас будет представлять интерес главным образом значение этого многочлена $\Phi_d(2)$ при $x = 2$, называемое *круговым числом* порядка d. Круговые числа играют важную роль при разложении на сомножители целых чисел вида $2^n - 1$.

Приведем два свойства круговых чисел и примеры таких чисел:

1) Если p — простое число, не являющееся делителем d, то необходимым и достаточным условием того, что $\Phi_d(2)$ имеет в качестве простого сомножителя число p, является выполнение равенства $E(2, p) = d$. Кроме того, p^β является делителем $\Phi_d(2)$ тогда и только тогда, когда p^β является делителем $2^d - 1$.

2) Пусть p — простое число, являющееся делителем d, а именно пусть $d = p^q d_1$ (p не делит d_1). Тогда необходимым и достаточным условием того, что $\Phi_d(2)$ имеет в качестве простого сомножителя число p, является выполнение равенства $E(2, p) = d_1$. При этом $\Phi_d(2)$ делится на p, но не делится на p^2.

[1]) Пронумерованные, как в (8.27). — *Прим. перев.*

[2]) Заметим, что $\Phi_d(x)$ — многочлен с целыми коэффициентами. — *Прим. ред.*

Для примера рассмотрим разложение числа $2^{18}-1$ на круговые числа:

$$x^{18}-1=\prod_{d\mid 18}\Phi_d(x),$$

$$2^{18}-1=\Phi_{18}(2)\cdot\Phi_9(2)\cdot\Phi_6(2)\cdot\Phi_3(2)\cdot\Phi_2(2)\cdot\Phi_1(2)=$$
$$=(3\cdot 19)\cdot(73)\cdot(3)\cdot(7)\cdot(3)\cdot(1).$$

Примеры круговых чисел приведены в табл. 8.7.

Таблица 8.7

Примеры круговых чисел

d	$\Phi_d(2)$	d	$\Phi_d(2)$	d	$\Phi_d(2)$	d	$\Phi_d(2)$	d	$\Phi_d(2)$
1	1	5	31	9	73	13	8 191	17	131 071
2	3	6	3	10	11	14	43	18	3×19
3	7	7	127	11	23×89	15	151	19	524 287
4	5	8	17	12	13	16	257	20	5×41

8.1.11. Минимальное представление дроби

Как известно, любую дробь $\alpha/B\,(0<\alpha<B)$ можно представить в виде

$$0,\ a_1a_2a_3\ldots.$$

Пусть B — произвольное заданное целое число, и пусть n — показатель числа r по модулю B. Тогда

$$r^n\equiv 1 \pmod B,$$

т. е. можно подобрать такое целое число A, что

$$r^n-1=AB.$$

Следовательно, для любого $\alpha<B$

$$\frac{\alpha}{B}=\alpha A\left(\frac{1}{r^n-1}\right)=\alpha A\,(0,\underbrace{\dot 0\dot 0\ldots 0\dot 1}_{\text{период } n})\text{ [1]}.$$

Это равенство показывает, что дробная часть является периодической последовательностью с периодом n.

Например, дробь $1/13=\dot 0{,}076923$ является десятичной последовательностью с периодом 6. Это следует из того, что $E(10,\ 13)=6$ и $10^6\equiv 1 \pmod{13}$. При этом $10/13=0{,}769230$; $100/13=7+9/13=7{,}692307$, $9/13=0{,}692307$; $1000/13=76+$

[1]) Точки над цифрой означают периодичность. — *Прим. ред.*

$+ 12/13 = 76{,}\dot{9}23076\dot{9}$; $12/13 = 0{,}\dot{9}23076\dot{9}$; $10\,000/13 = 769 + 3/13$, $3/13 = 0{,}\dot{2}30769\dot{2}$; $100\,000/13 = 7692 + 4/13$; $4/13 = 0{,}\dot{3}07692\dot{3}$. Заметим, что в данном примере дробные части 1/13, 10/13, 9/13, 12/13, 3/13, 4/13 получаются циклическими перестановками 076923.

В общем случае, если взять B так, что $E(r, B) = B - 1$, а именно чтобы число r было примитивным корнем B, то представление в виде десятичной дроби числа α/B $(0 < \alpha < B)$ получается путем циклических перестановок символов в представлении в виде десятичной дроби числа $1/B$. Это широко применяется в теории циклических AN-кодов и ниже будет использовано в частном случае $r = 2$.

8.2. Определение AN-кода

AN-код представляет собой отображение целых чисел 0, 1, 2, ..., N_0 в целые числа 0, $A \cdot 1$, $A \cdot 2$, ..., $A \cdot N_0$, где A — некоторое фиксированное для каждого кода целое положительное число, называемое порождающим числом. Числа 0, $A \cdot 1$, ..., A_0 называют кодовыми числами, а их представления в системе счисления по основанию r — кодовыми словами. В этой главе рассматривается случай двоичной системы счисления, когда $r = 2$, хотя многие результаты легко обобщаются и на случай произвольного r.

Очень важным свойством AN-кодов является то, что сумма кодовых чисел также является кодовым числом, а именно для любых двух целых чисел N_1 и N_2

$$A(N_1 + N_2) = AN_1 + AN_2. \tag{8.31}$$

Следовательно, если $N_1 + N_2 \leqslant N_0$, то число $A(N_1 + N_2) = AN_1 + AN_2$ также является одним из кодовых чисел. Поэтому, если закодированные числа сложить с помощью обычного сумматора, то полученная в результате сумма будет кодовым числом суммы исходных чисел. Благодаря этому AN-коды могут обнаруживать и исправлять ошибки, возникающие в сумматорах.

Если r является делителем A, т. е. если r является делителем каждого кодового числа AN, то в представлении чисел по основанию r самый младший разряд всегда будет равен 0. Далее, если A и r не являются взаимно простыми, то в младших разрядах, начиная с некоторого, будут появляться только определенные символы, тогда как другие цифры никогда не появятся. Поскольку это нежелательно, будем предполагать, что A и r взаимно просты. В случае $r = 2$ это означает, что число A

должно быть нечетно. AN-коды с $r = 2$ называются двоичными AN-кодами [1]).

Пример 8.1. AN-код с параметрами $A = 13$, $N_0 = 4$ и $r = 2$, т. е. двоичный $13N$-код, приведен в табл. 8.8.

Таблица 8.8

Двоичный 13 N-код

N	AN	Кодовые слова
0	0	000000
1	13	001101
2	26	011010
3	39	100111
4	52	110100

Например, сумме $1 + 2 = 3$ соответствуют следующие суммы кодовых чисел и кодовых слов этого кода: $13 + 26 = 39$ и $001101 + 011010 = 100111$. Если при переносе 1 из разряда 2^3 в разряд 2^4 происходит ошибка, в результате которой эта единица переноса превращается в 0, то результатом вычисления будет число $(010111)_2 = (23)_{10}$ (нижний индекс показывает основание системы счисления). Остаток от деления этого числа на 13 равен 10, а это означает, что полученная сумма не является кодовым числом $13N$-кода. Следовательно, возникшая ошибка будет всегда обнаружена. В действительности, как мы покажем ниже, эта ошибка может быть и исправлена (см. пример 8.6).

Так как максимальное кодовое число равно AN_0, то для представления чисел 0, $A \cdot 1$, $A \cdot 2$, ..., $A \cdot N_0$ в двоичной системе счисления требуется n двоичных символов, причем

$$2^{n-1} \leqslant AN_0 < 2^n,$$

т. е.

$$n = [(\log_2 AN_0) + 1]. \tag{8.32}$$

(Здесь $[x]$ — наибольшее целое число, не превосходящее x, или, другими словами, целая часть x [2]).) Это число n называют длиной двоичного AN-кода. Для представления исходных некодированных чисел 0 ,1, 2, ..., N_0 в двоичной системе счисления

[1]) Ниже, если это специально не оговорено, рассматриваются двоичные AN-коды.

[2]) Для любых двух действительных чисел x и y имеют место следующие неравенства: а) $x - 1 < [x] \leqslant x < [x] + 1$ и б) $[x] + [y] \leqslant [x+y] \leqslant [x] + [y] + 1$.

достаточно иметь

$$k = [\log_2 N_0 + 1] \tag{8.33}$$

двоичных знаков. Таким образом, разность

$$r = n - k \tag{8.34}$$

представляет собой число избыточных двоичных символов, необходимых для того, чтобы множество целых чисел $\{0, 1, 2, \ldots, N_0\}$ можно было представить с помощью AN-кода. Из формул (8.32)—(8.34) получаем

$$\log_2 A - 1 < r < \log_2 A + 1. \tag{8.35}$$

Следовательно, величина $\log_2 A$ является мерой избыточности AN-кода.

Пример 8.2. Избыточность $29N$-кода равна

$$\log_2 29 = 4{,}8.$$

Например, для представления целых чисел от 0 до $2^9 = 512$ включительно в двоичной системе счисления необходимо 9 разрядов, а для их представления в виде кодовых слов $29N$-кода требуется уже 14 двоичных символов. Следовательно, фактически необходимая избыточность равна пяти двоичным символам.

Избыточность $3N$-кода равна

$$\log_2 3 = 1{,}6.$$

Если этот код используется для представления десятичных цифр 0, 1, 2, ..., 9, то избыточность этого кода будет равна 1, поскольку в данном случае длина кода равна 5, а минимальное число двоичных символов, необходимых для представления тех же цифр в двоичной системе счисления, равно 4.

8.3. Арифметический вес и арифметическое расстояние

Определим сначала *арифметический вес* $W(N)$ произвольного целого числа N. Арифметический вес является одним из основных понятий в теории арифметических кодов и играет ту же роль, что и вес Хэмминга в теории алгебраических кодов.

Целое число N в двоичной системе счисления всегда может быть представлено единственным образом в следующем виде:

$$N = a_0 2^0 + a_1 2^1 + a_2 2^2 + \ldots + a_i 2^i + \ldots, \quad a_i \in \{0, 1\}, \tag{8.36}$$

$$i = 0, 1, 2, \ldots.$$

Если разрешить коэффициентам в этом представлении принимать значения 0, 1 и —1, то целое число N может быть также представлено в другом виде:

$$N = b_0 2^0 + b_1 2^1 + b_2 2^2 + \ldots + b_i 2^i + \ldots, \quad b_i \in \{-1, 0, +1\}, \quad i = 0, 1, 2, \ldots \tag{8.37}$$

Например, $(1111)_2 = 2^3 + 2^2 + 2^1 + 2^0$, и в то же время $(1111)_2 = 2^4 - 2^0$. Этот пример показывает, что второе из указанных выше представлений может иметь меньшее число ненулевых коэффициентов. Представление (8.37), имеющее наименьшее число ненулевых коэффициентов, называется *представлением с минимальным весом* или просто *минимальным представлением.*

Число членов с ненулевыми коэффициентами в минимальном представлении числа N называется арифметическим весом $W(N)$ числа N, т. е.

$$W(N) = \left[\begin{array}{l}\text{Число ненулевых коэффициентов} \\ \text{в минимальном представлении } N\end{array}\right]. \tag{8.38}$$

Ниже в данной главе арифметический вес будем называть просто весом.

В отличие от обычного двоичного представления в минимальном представлении коэффициенты могут принимать три значения $+1$, 0 и -1. Не следует, однако, смешивать минимальное представление с представлением в троичной системе счисления.

Кроме того, следует заметить, что минимальное представление не однозначно. Например,

$$43 = 101011 = 110\bar{1}0\bar{1} = 10\bar{1}0\bar{1}01,$$

где через $\bar{1}$ обозначены символы -1. Имеет место следующая лемма.

Лемма 8.1. *Если целое число N удовлетворяет условию $|N| \leqslant 2^m$, то во всех минимальных представлениях вида* (8.37) *числа N*

$$b_j = 0, \quad j \geqslant m + 1. \tag{8.39}$$

Доказательство. Предположим, что представление

$$b_0 2^0 + b_1 2^1 + b_2 2^2 + \ldots b_L 2^L, \quad b_L \neq 0,$$

числа N является минимальным представлением. Если $b_{L-1} = 0$, то

$$2^m \geqslant |N| = |b_0 2^0 + b_1 2^1 + \ldots + b_{L-2} 2^{L-2} + b_L 2^L| > 2^{L-1}.$$

Кроме того, если $b_{L-1} = b_L$, то очевидно, что

$$2^m \geqslant |N| = |b_0 2^0 + b_1 2^1 + \ldots + b_{L-2} 2^{L-2} + b_L 2^{L-1} + b_L 2^L| > 2^{L-1}.$$

Следовательно, в любом из этих случаев

$$2^m > 2^{L-1}.$$

Это означает, что $m > L - 1$, т. е. $m + 1 > L$.

При этом, если $b_{L-1} = -b_L$, то

$$b_{L-1} 2^{L-1} + b_L 2^L = (b_{L-1} + 2b_L) 2^{L-1} = b_L 2^{L-1}$$

и, следовательно, два слагаемых представления можно заменить одним. Это означает, что такое представление не является минимальным.

Из доказательства леммы 8.1 следует, что если в минимальном представлении произвольного числа N два соседних коэффициента b_j и b_{j+1} отличны от нуля, то $b_j = b_{j+1}$.

Еще одно важное свойство арифметического расстояния состоит в том, что для произвольных целых чисел N_1 и N_2

$$W(N_1 \pm N_2) \leqslant W(N_1) + W(N_2). \tag{8.40}$$

Действительно, если ненулевые коэффициенты при некоторой степени основания в минимальных представлениях чисел N_1 и N_2 отличны от нуля, то при сложении N_1 и N_2 они могут либо уничтожиться, либо может возникнуть перенос. В силу этого число ненулевых коэффициентов в сумме может быть лишь меньше суммы чисел ненулевых коэффициентов исходных минимальных представлений.

Пример 8.3. Так как $15 = 2^4 - 2^0$, то арифметический вес $W(15)$ числа 15 равен 2. Кодовым словом, соответствующим этому числу, является последовательность 1111, вес Хэмминга которой равен 4.

Арифметическое расстояние между целыми числами N_1 и N_2 определяется как арифметический вес абсолютного значения их разности и обозначается через $d(N_1, N_2)$. А именно

$$d(N_1, N_2) = W(|N_1 - N_2|). \tag{8.41}$$

Отсюда непосредственно следует, что

$$d(N_1, N_2) \geqslant 0. \tag{8.42}$$

Равенство в (8.42) имеет место только при $N_1 = N_2$. Заметим также, что

$$d(N_1, N_2) = d(N_2, N_1). \tag{8.43}$$

Для введенного расстояния выполняется также неравенство треугольника

$$d(N_1, N_3) \leqslant d(N_1, N_2) + d(N_2, N_3). \tag{8.44}$$

Действительно,

$$d(N_1, N_3) = W(|N_1 - N_2 + N_2 - N_3|) \leqslant$$

$$\leqslant W(|N_1 - N_2|) + W(N_2 - N_3|) = d(N_1, N_2) + d(N_2, N_3).$$

Как видно из формул (8.41)—(8.43), $d(N_1, N_2)$ определяет расстояние в произвольном множестве целых чисел. В этой и следующей главах арифметическое расстояние будем называть просто расстоянием.

Введенные выше арифметические вес и расстояние в теории арифметических кодов аналогичны весу и расстоянию Хэмминга, используемым в теории алгебраических кодов. Это связано с тем, что если при сложении двух чисел в двоичном представлении возникает ошибка в некотором разряде, в результате которой символ 0 переходит в 1 (или, наоборот, 1 переходит в 0), или ошибка переноса, то полученная в результате сумма будет отличаться от правильной суммы на величину $+2^j$ или -2^j, где j — некоторое целое число. В результате этой ошибки в двоичном представлении искомой суммы могут оказаться искаженными несколько разрядов и расстояние Хэмминга между этим представлением и правильным представлением будет больше 1, хотя арифметическое расстояние между правильной и неправильной суммами равно в точности 1. Другими словами, для арифметических кодов арифметические вес и расстояние более удобны по следующим двум причинам:

1) В арифметических устройствах ошибка из-за переноса может привести к искажению нескольких последующих разрядов. В арифметических кодах в отличие от алгебраических кодов ошибки в различных разрядах не являются независимыми.

2) В сумматорах и устройствах, осуществляющих вычитание, ошибки бывают двух типов; с переходами 1 в 0 и переходами 0 в 1 нельзя обращаться совершенно одинаково.

Эти причины хорошо иллюстрирует следующий пример.

Пример 8.4. Пусть $N_1 = 31$ и $N_2 = 32$. Эти числа имеют следующие двоичные представления: $N_1 = 31 = 011111$ и $N_2 = 32 = 100000$. Расстояние Хэмминга между этими последовательностями равно 6. Если в результате арифметической ошибки в разряде 2^0 к N_1 прибавляется 2^0 (конфигурация ошибки 000001), то $N_1 = 011111$ перейдет в $N_2 = 100000$. Поскольку $31 = 2^5 - 2^0$ и $32 = 2^5$, то, как следует из приведенных выше

определений арифметического веса и арифметического расстояния, $W(31) = 2$, $W(32) = 1$ и

$$d(31, 32) = W(32-31) = W(1) = 1,$$

что соответствует ошибке в одном разряде.

Если в результате арифметической ошибки в разряде 2^2 из N_2 вычитается 2^2 (конфигурация ошибки $000\bar{1}00$), то $N_2 = 32 = 100000$ переходит в $28 = 011100$. Расстояние Хэмминга между двумя последними последовательностями равно 4, а арифметическое расстояние между 32 и 28 равно

$$d(32, 28) = W(32-28) = W(4) = 1.$$

Это соответствует ошибочному вычитанию 2^2.

Рассмотрим, далее, методы нахождения минимального представления числа N. Как уже указывалось выше, минимальное представление не единственно. Однако известно, что если в представлении (8.37) никакие два соседних коэффициента одновременно не являются ненулевыми, т. е.

$$b_i b_{i+1} = 0, \quad i = 0, 1, 2, \ldots \tag{8.45}$$

(ниже это условие называется условием М), то представление (8.37) является минимальным и, более того, единственно [8].

Минимальность и *единственность* представления, удовлетворяющего условию М, может быть доказана следующим образом.

Предположим, что представление (8.37) целого числа N:

$$N = b_0 2^0 + b_1 2^1 + \ldots + b_n 2^n = \sum_{i=0}^{n} b_i 2^i, \quad b_i \in \{0, +1, -1\}, \tag{1}$$

удовлетворяет условию М, а именно

$$b_i b_{i+1} = 0, \quad i = 0, 1, \ldots, n-1. \tag{2}$$

Сначала докажем единственность представления вида (1) заданного целого числа N. Допустим, что существуют два различных представления числа N в виде (1), т. е.

$$N = \sum_{i=0}^{n} b'_i 2^i = \sum_{i=0}^{n} b''_i 2^i, \quad b'_i, \ b''_i \in \{0, +1, -1\}, \tag{3}$$

и придем к противоречию. Действительно, пусть m — минимальный из индексов i, таких, что $b'_i \neq b''_i$. Тогда

$$b'_m - b''_m = \pm(1 + \alpha), \tag{4}$$

где $\alpha = 0$ или 1, и

$$\sum_{i=0}^{m} b'_i 2^i - \sum_{i=0}^{m} b''_i 2^i = \pm(1 + \alpha) 2^m. \tag{5}$$

Если $m = n$, равенство (3) не может выполняться. При $m < n$ и $\alpha = 1$ из равенства (4) следует, что $b'_m \neq 0 \neq b''_m$. Но тогда, согласно формуле (2), $b'_{m+1} = 0 = b''_{m+1}$. Из равенства (5) следует, что в этом случае

$$\sum_{i=0}^{m+\alpha} b'_i 2^i - \sum_{i=0}^{m+\alpha} b''_i 2^i = \pm (1 + \alpha) 2^m. \tag{6}$$

Очевидно, что при $m < n$ и $\alpha = 0$ равенство (6) можно получить непосредственно из (5). Таким образом, при $m < n$ равенство (6) справедливо как при $\alpha = 0$, так и при $\alpha = 1$. При $m + \alpha \geqslant n$ получаем противоречие с равенством (3). Рассмотрим случай $m + \alpha < n$. Равенства (3) и (6) могут выполняться одновременно только в том случае, если

$$\sum_{i=m+\alpha+1}^{n} b'_i 2^i - \sum_{i=m+\alpha+1}^{n} b''_i 2^i = \mp (1 + \alpha) 2^m = \mp 2^{m+\alpha}, \tag{7}$$

но это невозможно, поскольку левая часть последнего равенства является целым положительным кратным $2^{m+\alpha+1}$, тогда как правая часть равна $\mp 2^{m+\alpha}$. Таким образом, если представление, удовлетворяющее условию М, существует, то оно единственно.

Далее для произвольного целого числа N докажем существование представления, удовлетворяющего условию М, а также минимальность числа

$$T_n = \sum_{i=0}^{n} |b_i| \tag{8}$$

ненулевых коэффициентов в нем.

Представим число N в виде (1):

$$N = \sum_{i=0}^{n} b_i^{(v)} 2^i, \quad n \geqslant v \geqslant 0, \tag{9}$$

и положим

$$T_n^{(v)} = \sum_{i=0}^{n} \left| b_i^{(v)} \right|. \tag{10}$$

При $n = 0$ существование и минимальность этого представления очевидны.

Рассмотрим случай $n > 0$. Предположим, что $\left\{b_i^{(v)}\right\}$ — совокупность коэффициентов $b_i^{(v)}$ $(i = 0, 1, \ldots, n;\ n > v \geqslant 0)$, которые удовлетворяют условию М до разряда 2^v, и найдем совокупность $\left\{b_i^{(v+1)}\right\}$ коэффициентов представления (9) числа N, которые удовлетворяют условию М до разряда 2^{v+1}. Совокупность $\left\{b_i^{(v)}\right\}$ удовлетворяет одному из следующих трех взаимоисключающих условий:

$$b_{v+1}^{(v)} b_v^{(v)} = +1, \tag{11а}$$

$$b_{v+1}^{(v)} b_v^{(v)} = 0, \tag{11б}$$

$$b_{v+1}^{(v)} b_v^{(v)} = -1. \tag{11в}$$

При выполнении равенств (11а) и (11в) имеют место также следующие два соотношения:

$$b_u^{(v)} \neq b_{u-1}^{(v)} = b_{u-2}^{(v)} = \ldots = b_v^{(v)} = \pm 1, \quad n \geqslant u \geqslant v+2, \tag{12а}$$

$$\pm 1 = b_{v+1}^{(v)} \neq b_v^{(v)} = \pm 1. \tag{12в}$$

Следовательно,

$$\sum_{i=v}^{u} b_i^{(v)} 2^i = \left(b_u^{(v)} + b_v^{(v)}\right) 2^u - b_v^{(v)} 2^v, \tag{13а}$$

$$\sum_{i=v}^{v+1} b_i^{(v)} 2^i = 0 \cdot 2^{v+1} - b_v^{(v)} 2^v, \tag{13в}$$

где u — индекс, значения которого лежат в области $n \geqslant u \geqslant v+2$. Для каждого из трех возможных случаев, определяемых равенствами (11а), (11б) и (11в), соответственно по формулам (11б), (13а) и (13в) находим

$$\left.\begin{aligned} b_u^{(v+1)} &= b_u^{(v)} + b_v^{(v)}, \\ b_i^{(v+1)} &= 0, \quad u > i > v, \\ b_v^{(v+1)} &= -b_v^{(v)}, \\ b_i^{(v+1)} &= b_i^{(v)}, \end{aligned}\right\} \tag{14а}$$

$$b_i^{(v+1)} = b_i^{(v)}, \quad n \geqslant i \geqslant 0, \tag{14б}$$

$$\left.\begin{aligned} b_{v+1}^{(v+1)} &= 0, \\ b_v^{(v+1)} &= -b_v^{(v)}, \\ b_i^{(v+1)} &= b_i^{(v)}. \end{aligned}\right\} \tag{14в}$$

Очевидно, что при данном методе построения получающаяся совокупность $\left\{b_i^{(v+1)}\right\}$ удовлетворяет условию М до разряда 2^{v+1} и при этом

$$T_n^{(v+1)} \leqslant T_n^{(v)}. \tag{15}$$

Таким образом, произвольное заданное представление (9) при $v = 0$ удовлетворяет условию М до разряда 2^0. Далее, взяв за основу множество $\left\{b_i^{(0)}\right\}$ и строя с помощью описанной выше процедуры последовательно множества $\left\{b_i^{(1)}\right\}, \ldots, \left\{b_i^{(n)}\right\}$, можно найти $\left\{b_i^{(n)}\right\}$. При этом

$$T_n^{(0)} \geqslant T_n^{(1)} \geqslant \ldots \geqslant T_n^{(n)}, \tag{16}$$

т. е. $T_n^{(n)}$ меньше соответствующего параметра любого другого множества $\{b_i\}$.

Алгоритм нахождения представления, удовлетворяющего условию М

Рассмотрим простой алгоритм нахождения представления, удовлетворяющего условию *М*, по обычному представлению целого числа *N* в двоичной системе счисления. Пусть

$$N = a_{n-1}a_{n-2} \ldots a_1a_0, \quad a_i \in \{0, 1\}, \quad a_{n-1} \neq 0 \tag{8.46}$$

и

$$N = \sum_{i=0}^{n-1} a_i 2^i. \tag{8.47}$$

Так как $3N = 2N + N$, то, сдвигая правую часть (8.46) на 1 разряд влево и прибавляя N, получаем

$$3N = b_{n+1} b_n b_{n-1} \ldots b_1 b_0, \quad b_i \in \{0, 1\}, \tag{8.48}$$

где

$$\left.\begin{aligned} &b_0 = a_0 \text{ — коэффициент при } 2^0 \text{ в представлениях } N \\ &\qquad \text{и } 3N, \\ &b_i \equiv a_i + a_{i-1} + c_i \pmod 2, \quad 1 \leqslant i \leqslant n-1, \\ &c_i \quad \text{равен 1 при наличии переноса из } (i-1)\text{-го раз-} \\ &\qquad \text{ряда и 0 в остальных случаях,} \\ &b_n \equiv a_{n-1} + c_n, \\ &b_{n+1} \equiv c_{n+1}. \end{aligned}\right\} \tag{8.49}$$

Далее, пользуясь тем, что $2N = 3N - N$, и вычитая (8.46) поразрядно из (8.48), получаем

$$\begin{array}{r} b_{n+1}\ b_n\ b_{n-1}\ b_{n-2} \ldots b_i \ldots b_1\ b_0 \\ -\quad\quad a_{n-1}\ a_{n-2} \ldots a_i \ldots a_1\ a_0 \\ \hline d_{n+1}\ d_n\ d_{n-1}\ d_{n-2} \ldots d_i \ldots d_1\ d_0 \end{array}, \tag{8.50}$$

где

$$d_i = \begin{cases} 1, & b_i > a_i, \\ -1, & b_i < a_i, \\ 0, & b_i = a_i. \end{cases} \tag{8.51}$$

Лемма 8.2. *Представление* $d_{n+1} d_n \ldots d_i \ldots d_1 d_0$ *числа* $2N$, *задаваемое формулами* (8.50) *и* (8.51), *удовлетворяет условию* М.

Доказательство. Очевидно, что $d_0 = 0$. Следовательно, для доказательства леммы достаточно показать, что если $d_i \neq 0$ при некотором $1 \leqslant i \leqslant n$, то $d_{i+1} = 0$. Предположим, что $d_i \neq 0$. Пусть $a_i = 1$ и $b_i = 0$ (при $a_i = 0$ и $b_i = 1$ доказательство проводится аналогично). Тогда

$$b_i \equiv a_i + a_{i-1} + c_i \equiv 1 + a_{i-1} + c_i \equiv 0 \pmod 2.$$

Так как $1 \geqslant a_{i-1} \geqslant 0$, $1 \geqslant c_i \geqslant 0$, то $c_{i+1} = 1$. Следовательно,

$$b_{i+1} \equiv a_{i+1} + a_i + c_{i+1} \equiv a_{i+1} + 1 + 1 \equiv a_{i+1} \pmod 2,$$

так что

$$d_{i+1} = 0.$$

Таким образом, как утверждается в лемме 8.2, представление $d_{n+1}d_n \ldots d_1d_0$ числа $2N$ удовлетворяет условию М и, следовательно, является минимальным. Число ненулевых слагаемых в этом представлении $\sum_{i=0}^{n+1} |d_i|$ является весом $W(2N)$ числа $2N$, который равен весу $W(N)$ числа N. Действительно, так как $2N$ является результатом деления $2N$ на 2, то последовательность, полученная сдвигом представления $2N$ на один разряд вправо, будет представлением N. Поскольку $d_0 = 0$, то в результате этого преобразования вес не изменяется.

Из этой леммы, кроме всего прочего, непосредственно следует, что 1) число разрядов в минимальном представлении числа, обычное двоичное представление которого имеет n разрядов, не превышает $n + 1$ и 2) вес двоичного n-разрядного числа не превышает $[n/2] + 1$.

Пример 8.5. Используя лемму 8.2, найдем вес числа $N = 3673$. Так как $N = 0111001011001$ и $3N = 10101100001011$, то

$$\begin{array}{r} 3N = 10101100001011 \\ -\ N = \ \ 0111001011001 \\ \hline 2N = 100\bar{1}010\bar{1}0\bar{1}0010 \\ N = \ \ 100\bar{1}010\bar{1}0\bar{1}001 \end{array}$$

и, следовательно, $W(N) = 6$.

8.4. Минимальное расстояние и корректирующая способность *AN*-кода

Если в процессе передачи или некоторых вычислений целое число N в результате ошибок переходит в число N' и расстояние $d(N, N') = W(|N - N'|)$ между N и N' равно t, то будем говорить, что произошла *ошибка веса t*.

Пусть I — целое число, совпадающее с кодовым числом AN при отсутствии ошибок и, быть может, отличающееся от AN при наличии ошибок. Например, I может быть выходом сумматора, суммы на выходе которого в отсутствие ошибок являются кодовыми числами AN-кода. В этом случае

$$I = AN + E, \tag{8.52}$$

где E — целое число, называемое ошибкой. Если $W(E) = t$, то E называется ошибкой веса t.

Минимальное расстояние $d_{\text{мин}}$ кода — это по определению минимальное значение расстояния между различными кодовыми числами. Точнее, минимальное расстояние AN-кода, который со-

поставляет числам 0, 1, ..., N_0 кодовые числа 0, $A\cdot 1$, ..., $A\cdot N_0$, определяется следующим образом:

$$d_{\text{мин}} = \min_{N_1 \neq N_2} W(AN_1, AN_2), \quad 0 \leqslant N_1, \quad N_2 \leqslant N_0. \tag{8.53}$$

Теорема 8.1. *AN-код исправляет все ошибки веса t и менее тогда и только тогда, когда его минимальное расстояние $d_{\text{мин}}$ не меньше чем $2t+1$. AN-код может обнаруживать все ошибки веса d и менее тогда и только тогда, когда его минимальное расстояние не меньше чем $d+1$.*

Доказательство. Если ни для каких двух различных кодовых чисел AN_1 и AN_2 не существует пары ошибок E_1 и E_2 веса t или менее, такой, что

$$AN_1 + E_1 = AN_2 + E_2, \tag{8.54}$$

то по $I_1 = AN_1 + E_1$ всегда можно однозначно восстановить AN_1. Верно и обратное; в самом деле, из формул (8.54) и (8.40) имеем

$$d(AN_1, AN_2) = W(|AN_1 - AN_2|) = \\ = W(|E_2 - E_1|) \leqslant W(E_1) + W(E_2) \leqslant 2t. \tag{8.55}$$

Следовательно, условие $d_{\text{мин}} \geqslant 2t+1$ является достаточным для того, чтобы AN-код мог исправлять все ошибки веса t и менее. Наоборот, если допустить, что $d(AN_1, AN_2) \leqslant 2t$, то число $AN_1 - AN_2$ будет иметь минимальное представление веса $2t$ или менее. Пусть E_1 — сумма t ненулевых членов высших разрядов этого минимального представления, а E_2 — сумма оставшихся членов этого представления. Вес определенных таким образом ошибок E_1 и E_2 не превосходит t, и E_1, E_2 удовлетворяют равенству (8.54). Следовательно, они не могут быть исправлены.

Далее, AN-код может обнаруживать все ошибки веса d и менее тогда и только тогда, когда ни для каких двух различных кодовых слов AN_1 и AN_2 не существует ни одной ошибки веса d и менее, такой, что

$$AN_1 + E = AN_2. \tag{8.56}$$

Вторая часть теоремы доказывается точно так же, как и первая.

Аналогично можно доказать, что AN-код может исправлять ошибки веса t и менее и одновременно обнаруживать ошибки веса $d(>t)$ и менее тогда и только тогда, когда его минимальное расстояние не меньше чем $t+d+1$.

Эти соотношения между коорректирующей способностью и минимальным расстоянием AN-кодов совершенно аналогичны соответствующим соотношениям для алгебраических кодов. Они

подтверждают сделанное в примере 8.4 и ранее утверждение о том, что арифметический вес, арифметическое расстояние и минимальное расстояние являются понятиями, которые играют в теории арифметических кодов очень важную роль.

Теорема 8.2. *Расстояние* $d(AN_1, AN_2)$ *между двумя произвольными различными кодовыми числами* AN_1 *и* AN_2 *AN-кода равно весу некоторого ненулевого кодового числа* AN_3. *Наоборот, для любого ненулевого кодового числа* AN_3 *веса* $W(AN_3)$ *найдутся такие два различных кодовых числа* AN_1 *и* AN_2, *что расстояние между ними будет в точности равно* $W(AN_3)$.

Доказательство. Пусть AN_1 и AN_2 — различные кодовые числа. Так как $0 \leqslant N_1 \leqslant N_0$ и $0 \leqslant N_2 \leqslant N_0$, то число $N_3 = |N_1 - N_2|$ также удовлетворяет неравенству $0 < N_3 \leqslant N_0$. Следовательно, AN_3 — ненулевое кодовое число и $W(AN_3) = d(AN_1, AN_2)$.

Наоборот, вес $W(AN_3)$ любого ненулевого кодового числа AN_3 совпадает с расстоянием между AN_3 и кодовым числом $A \cdot 0 = 0$.

Еще одной важной характеристикой AN-кодов является параметр $M(A, d)$, который определяется следующим образом:

$$M(A, d) = \begin{bmatrix}\text{минимальное целое число } m, \\ \text{такое, что вес } mA \text{ меньше } d\end{bmatrix}. \qquad (8.57)$$

Непосредственным следствием теоремы 8.2 является следующее утверждение.

Утверждение 8.2.1. AN-код $\{0, A \cdot 1, A \cdot 2, \ldots, A \cdot N_0\}$, для которого $N_0 = M(A, d) - 1$, имеет минимальное расстояние d.

Пример 8.6. Если $A = 13$ и $d = 3$, то $M(13, 5) = 5$. Действительно,

$$\begin{aligned}
13 \times 0 &= 0 = 0000000, & W(0) &= 0;\\
13 \times 1 &= 13 = 0010\bar{1}01, & W(13) &= 3;\\
13 \times 2 &= 26 = 010\bar{1}010, & W(26) &= 3;\\
13 \times 3 &= 39 = 0101001, & W(39) &= 3;\\
13 \times 4 &= 52 = 10\bar{1}0100, & W(52) &= 3;\\
13 \times 5 &= 65 = 1000001, & W(65) &= 2.
\end{aligned}$$

Таким образом, минимальное расстояние AN-кода, описанного в примере 8.1, равно 3, и, следовательно, согласно теореме 8.1, он может исправлять любые ошибки веса 1.

Как следует из утверждения 8.2.1, вычисление значений функции $M(A, d)$ является одной из важнейших проблем теории

Таблица 8.9

Примеры *AN*-кодов

AN-коды с минимальным расстоянием 3			*AN*-коды с минимальным расстоянием 4			*AN*-коды с минимальным расстоянием 5		
A	$M(A, 3)$	$AM(A, 3)$	A	$M(A, 4)$	$AM(A, 4)$	A	$M(A, 5)$	$AM(A, 5)$
11	3	$2^5 + 1$	43	3	$2^7 + 1$	171	3	$2^9 + 1$
13	5	$2^6 + 1$	51	5	$2^8 + 1$	205	5	$2^{10} + 1$
19	27	$2^9 + 1$	89	23	$2^{11} - 1$	557	11	$2^{12} + 2^{11} - 2^1 - 1$
23	89	$2^{11} - 1$	105	39	$2^{12} - 1$	565	29	$2^{14} + 1$
29	565	$2^{14} + 1$	153	215	$2^{15} + 2^7 - 1$	885	35	$2^{15} - 2^{11} + 2^8 - 1$
37	7 085	$2^{18} + 1$	185	1 417	$2^{18} + 1$	941	37	$2^{15} + 2^{11} + 1$
47	178 481	$2^{23} - 1$	267	15 709	$2^{22} - 1$	1 069	69	$2^{16} + 2^{13} + 2^5 + 1$
53	1 266 205	$2^{26} + 1$	351	23 905	$2^{23} + 2^{11} - 1$	1 417	185	$2^{18} + 1$
59	9 099 507	$2^{29} + 1$	353	25 249	$2^{23} + 2^{19} + 1$	1 883	211	$2^{19} - 2^{17} + 2^{12} + 1$
61	17 602 325	$2^{30} + 1$	357	46 995	$2^{24} - 1$	1 939	507	$2^{20} - 2^{16} + 2^5 + 1$
67	128 207 979	$2^{33} + 1$	393	181 433	$2^{26} + 2^{22} + 1$	2 399	881	$2^{21} + 2^{14} - 2^4 - 1$
71	483 939 977	$2^{35} - 1$	547	245 131	$2^{27} - 2^{17} + 1$	2 669	1 179	$2^{21} - 2^{20} + 2^{10} - 1$
79	6 958 934 353	$2^{39} - 1$	555	123 818 877	$2^{36} - 1$	2 807	2 987	$2^{23} - 2^{12} - 2^1 - 1$
83	26 494 256 091	$2^{51} + 1$	737	5 967 134 431	$2^{42} - 2^{28} - 1$			
101	5 099 523 830 125	$2^{60} + 1$	801	10 981 392 129	$2^{43} + 2^{21} - 1$			

арифметических кодов. Эта проблема полностью решена только при $d = 2, 3$. Соответствующее решение приводится в разд. 8.5. За исключением этого частного случая, общих методов вычисления $M(A, d)$, которые требовали бы небольшого числа операций, нет. В табл. 8.9 приведено несколько примеров AN-кодов.

8.5. Обнаружение и исправление независимых ошибок веса 1

Как следует из теоремы 8.1, код с минимальным расстоянием $d_{\text{мин}} = 2$ является кодом, обнаруживающим независимые ошибки веса 1. Следующая теорема дает простой способ построения таких кодов.

Теорема 8.3. *Если* $A(>1)$ *— нечетное число, то*

$$M(A, 2) = \infty. \tag{8.58}$$

Доказательство. Если допустить существование конечного целого положительного числа N, такого, что $W(AN) = 1$, то при некотором целом положительном j будет иметь место равенство $AN = 2^j$. Следовательно, A делится на 2, но этого не может быть, поскольку по условию теоремы A — нечетное число. Это доказывает (8.58).

Согласно этой теореме, при любом нечетном $A(>1)$ и любом сколь угодно большом N_0 минимальное расстояние AN-кода $\{0, A\cdot 1, A\cdot 2, A\cdot 3, \ldots, A\cdot N_0\}$ не меньше 2, и, следовательно, при любой кодовой длине такие AN-коды могут обнаруживать независимые ошибки веса 1. Среди этих кодов код с $A = 3$ имеет минимальную избыточность. Для этого кода $\log_2 A = \log_2 3 = 1{,}5849\ldots$ и число избыточных разрядов не провосходит 2.

Коды с минимальным расстоянием $d_{\text{мин}} = 3$ являются кодами, которые либо исправляют ошибки веса 1, либо обнаруживают ошибки веса 2. Как показывают две следующие теоремы, коды с минимальным расстоянием 3 также могут быть построены сравнительно просто.

В разд. 8.1.7 было дано определение показателя $e = E(a, m)$ целого числа a по модулю m. Для взаимно простых a и m введем величину $d = D(a, m)$ следующим образом. По определению $D(a, m)$ — минимальное целое положительное число d, такое, что

$$a^d \equiv -1 \pmod m. \tag{8.59}$$

Если $D(a, m)$ существует, то

$$2D(a, m) = E(a, m). \tag{8.60}$$

Последнее равенство следует из того, что если $a^d \equiv -1$, то $a^{2d} \equiv 1$ и ни при каком $e(<2d)$ сравнение $a^e \equiv 1$ выполняться не может.

Теорема 8.4. *Если $A(>1)$ — нечетное число, то*

$$M(A, 3) = \begin{cases} (2^e - 1)/A, & e = E(2, A), \quad \text{если } d = D(2, A) \\ & \text{не существует}, \\ (2^d + 1)/A, & d = D(2, A). \end{cases} \tag{8.61}$$

Доказательство. Сначала предположим, что $D(2, A)$ не существует. По определению e

$$2^e \equiv 1 \pmod{(A)},$$

или, что то же самое, $2^e - 1 \equiv 0 \pmod{A}$. Следовательно, существует целое положительное число N_1, такое, что $AN_1 = 2^e - 1$. Отсюда и из определения $M(A, 3)$ имеем

$$M(A, 3) \leqslant N_1 = \frac{2^e - 1}{A}. \tag{8.62}$$

Покажем, далее, что целые положительные числа N, такие, что $W(AN) \leqslant 2$, удовлетворяют неравенству $N \geqslant N_1$. Во-первых, очевидно, что $W(AN) \neq 0$. Далее, поскольку A — нечетно, то, согласно теореме 8.3, $W(AN) \neq 1$. Если же $W(AN) = 2$, то существуют такие целые числа i и j $(i > j \geqslant 0)$, что

$$AN = 2^i \pm 2^j = 2^j(2^{i-j} \pm 1).$$

Так как A — нечетно, то $(A, 2^j) = 1$ и, следовательно, A должно делить $2^{i-j} \pm 1$, т. е. $2^{i-j} \pm 1 \equiv 0 \pmod{A}$. Из определения $e \equiv E(2, A)$ имеем $i - j \geqslant e$. Следовательно [1]),

$$AN = 2^j(2^{i-j} - 1) \geqslant 2^e - 1 = AN_1,$$

и, таким образом, мы показали, что $N \geqslant N_1$.

Если $d = D(2, A)$ существует, то доказательство проводится аналогично.

Лемма 8.3. *AN-код $\{AN \mid 0 \leqslant N < M\}$ имеет минимальное расстояние $d_{\text{мин}} = 3$ тогда и только тогда, когда*

$$\alpha 2^i \not\equiv \beta 2^j \pmod{A} \tag{8.63}$$

для всех i и j $(i \neq j)$, таких, что $2^i + 2^j < AM$ и $\alpha, \beta \in \{+1, -1\}$.

Доказательство. $|\alpha 2^i - \beta 2^j| \leqslant |2^i + 2^j| < AM$. Так как, согласно формуле (8.63), $\alpha 2^i - \beta 2^j \not\equiv 0 \pmod{A}$, то в число кодовых чисел рассматриваемых AN-кодов не входят целые числа

[1]) По предположению $2^{i-j} + 1 \not\equiv 0 \pmod{A}$. — *Прим. перев.*

веса 2. Кроме того, если существует целое i, такое, что $2^i < AM$ и $\alpha 2^i \equiv 0 \pmod{A}$, то $A = 2^k$, $k|i$ $(k \leqslant i)$ и $|2^i - 2^k| < AM$. В этом случае $2^i - 2^k \equiv 0 \pmod{A}$, что противоречит условиям леммы. Следовательно, в число кодовых чисел рассматриваемых AN-кодов не входят и числа веса 1. Кроме того, заметим, что существует только одно число веса 0, а именно $A \cdot 0 = 0$. Следовательно, минимальное расстояние рассматриваемых кодов не меньше 3.

Предположим наоборот, что $\alpha 2^i \equiv \beta 2^j \pmod{A}$ для некоторых i и j $(i \neq j)$, таких, что $2^i + 2^j < AM$. Тогда, так как $D = |\alpha 2^i - \beta 2^j| < |2^i + 2^j| < AM$ и $|\alpha 2^i - \beta 2^j| \equiv 0 \pmod{A}$, то число D является кодовым числом рассматриваемого AN-кода, и, следовательно, последний имеет минимальное расстояние 2 или меньше.

Теорема 8.5. *Пусть $A(>1)$ — нечетное простое число. Тогда полная система вычетов по модулю A образует конечное поле $GF(A)$. Если 2 является примитивным элементом* [1]) *$GF(A)$, то*

$$M(A, 3) = \frac{2^{(A-1)/2} + 1}{A}. \tag{8.64а}$$

Если 2 не является примитивным элементом $GF(A)$, но — 2 является примитивным корнем $GF(A)$, то

$$M(A, 3) = \frac{2^{(A-1)/2} - 1}{A}. \tag{8.64б}$$

Верно и обратное, а именно если выполняется (8.64а), *то A — нечетное простое число и 2 — примитивный корень по модулю A. Далее, если выполняется равенство* (8.64б), *то A — нечетное простое число, —2 — примитивный корень, а 2 примитивным корнем не является.*

Доказательство. Так как A — нечетное простое число, то $(2, A) = 1$ и, согласно теореме Ферма,

$$2^{A-1} - 1 = (2^{(A-1)/2} + 1)(2^{(A-1)/2} - 1) \equiv 0 \pmod{A}. \tag{8.65}$$

В разд. 8.1.5 указывалось, что полная и приведенная системы вычетов по простому модулю A являются соответственно полем $GF(A)$ и его мультипликативной группой $G(A)$.

Если 2 является примитивным элементом $G(A)$, то $2^{(A-1)/2} \not\equiv 1 \pmod{A}$. В этом случае $2^{(A-1)/2} + 1 \equiv 0$, т. е. $2^{(A-1)/2} \equiv -1$, и так как 2 — примитивный элемент $G(A)$, то множество чисел $2^0 = 1, 2^1, 2^2, \ldots, 2^{(A-3)/2}, 2^{(A-1)/2} \equiv -1, -2^{-1}, -2^{-2}, \ldots, -2^{(A-3)/2}$ будет приведенной системой вычетов по модулю A. Следователь-

[1]) Точнее, предполагается, что 2 является образующей мультипликативной группы $G(A)$.

но числа $\pm 2^j$ ($j \leqslant (A-3)/2$) при делении на A имеют различные остатки. Замечая, что $2^{(A-1)/2}+1 \equiv 0 \pmod{A}$, и полагая $AM = 2^{(A-1)/2}+1$, получаем, что $\alpha 2^i \not\equiv \beta 2^j \pmod{A}$ для всех i и j, удовлетворяющих условию $|2^i + 2^j| < AM$ (здесь α, $\beta \in \{+1, -1\}$). Отсюда и из леммы 8.3 следует справедливость равенства (8.64а).

Далее рассмотрим случай, когда примитивным элементом $GF(A)$ является число -2. Если $(A-1)/2$ — четное число, то $2^{(A-1)/2} \equiv (-2)^{(A-1)/2} \not\equiv 1 \pmod{A}$, и, поскольку A — простое число, то, используя формулу (8.65), вновь приходим к сравнению

$$2^{(A-1)/2}+1 \equiv 0, \quad \text{т. е.} \quad 2^{(A-1)/2} \equiv -1.$$

Так как -2 является примитивным элементом, то числа

$$2^0, 2^1, 2^2, \ldots, 2^{(A-3)/2}, 2^{(A-1)/2} \equiv -1, -2^1, -2^2, \ldots, -2^{(A-3)/2}$$

принадлежат различным классам вычетов. Это означает, что число 2 также является примитивным корнем и, следовательно, в этом случае применима формула (8.64а).

Если -2 является примитивным корнем, а 2 не является примитивным корнем, то, как следует из вышеизложенного, число $(A-1)/2$ должно быть нечетным. Кроме того, $(-2)^{(A-1)/2} - 0 \not\equiv 0 \pmod{A}$. Так как $(-2)^{(A-1)/2} - 1 = -2^{(A-1)/2} - 1 \not\equiv \not\equiv 0 \pmod{A}$, то $2^{(A-1)/2}+1 \not\equiv 0 \pmod{A}$. Вновь обращаясь к формуле (8.65), получаем, что $2^{(A-1)/2} - 1 \equiv 0 \pmod{A}$, а именно $2^{(A-1)/2} \equiv 1 \pmod{A}$. Отсюда следует, что приведенной системой вычетов является также множество чисел

$$1, -2, 2^2, -2^3, \ldots, 2^{(A-3)/2} - 2^{(A-1)/2} = -1, 2, -2^2, \ldots, -2^{(A-3)/2}.$$

Таким образом, числа $\pm 2^j$ ($j \leqslant (A-3)/2$) являются попарно несравнимыми по модулю A. Так как $2^{(A-1)/2} - 1 \equiv 0 \pmod{A}$, то, полагая $AM = 2^{(A-1)/2} = 1$, получаем, что $\alpha 2^i \not\equiv \beta 2^j \pmod{A}$ для всех i и j ($i \neq j$), удовлетворяющих условию $|2^i + 2^j| < AM$. Отсюда из леммы 8.3 следует справедливость равенства (8.64б).

Наоборот, если имеют место равенства (8.64а) или (8.64б), то из леммы 8.3 следует, что $\alpha 2^i \not\equiv \beta 2^j \pmod{A}$ для всех i и j ($i \neq j$), удовлетворяющих соответственно условию

$$|2^i + 2^j| < 2^{(A-1)/2} + 1$$

или

$$|2^i + 2^j| < 2^{(A-1)/2} - 1,$$

и любых α, $\beta \in \{+1, -1\}$. Эти условия выполняются только в том случае, если $(A-1)/2$ чисел $\pm 2^i$ ($0 \leqslant i \leqslant (A-3)/2$), а именно

$$2^0, 2^1, 2^2, \ldots, 2^{(A-3)/2},$$

пробегают все элементы приведенной системы вычетов по модулю A. Следовательно, вычеты по модулю A имеют вид $\pm 2^i$ и A является нечетным числом. Отсюда в свою очередь следует, что число A взаимно просто с числом 2 и, кроме того, со всеми вычетами по модулю A, т. е. A является нечетным простым числом. Это означает, что система вычетов по модулю A является конечным полем. Так как все степени числа 2, не превосходящие $(A-1)/2$, различны, то порядок числа 2 не меньше $(A-1)/2$. Кроме того, заметим, что порядок числа 2 должен делить $A-1$. Следовательно, порядок числа 2 равен либо $(A-1)/2$, либо $A-1$. Для числа -2 рассуждения полностью аналогичны. В любом конечном поле существует примитивный элемент a [1]). Если порядки чисел -2 и 2 равны $(A-1)/2$, то числа вида $\pm 2^j$ должны быть нечетными степенями a, но этого быть не может. Следовательно, по крайней мере одно из чисел 2 или -2 имеет порядок $A-1$, т. е. одно из чисел 2 или -2 является примитивным корнем.

По аналогии с совершенными кодами Хэмминга, исправляющими ошибки веса 1, даваемые теоремой 8.5 AN-коды, исправляющие ошибки веса 1, также можно назвать совершенными в указанном ниже смысле. AN-код длины n, исправляющий ошибки веса 1, должен обладать способностью различать посредством вычетов по модулю A каждую из $2m$ возможных ошибок $\pm 2^0, \pm 2^1, \ldots, \pm 2^{m-1}$ веса 1 и правильное кодовое число, т. е. всего $2m+1$ различных исходов. Согласно лемме 8.3, каждому из этих $2m+1$ исходов должен соответствовать «собственный» вычет, что возможно только в случае, если $A \geqslant 2m+1$. Теорема 8.5 дает необходимые и достаточные условия для того, чтобы выполнялось точное равенство. Условия теоремы таковы, что у даваемых ею кодов число A вычетов $0, 1, 2, \ldots, A-1$ по модулю A равно $2m+1$ возможным различным исходам. В этом смысле эти коды являются совершенными. Все AN-коды с минимальным расстоянием $d_{\text{мин}} = 3$, приведенные в табл. 8.9, являются кодами этого типа.

Чанг и Рид [12] доказали следующую теорему, из которой следует, что, за исключением кодов, указанных в теореме 8.5, других совершенных AN-кодов, исправляющих ошибки веса 1, не существует.

Теорема 8.6. *AN-код длины n, исправляющий ошибки веса 1, является совершенным тогда и только тогда, когда A — простое число, $A = 2m+1$ и одно из чисел 2 или -2 является примитивным корнем по модулю A.*

---

[1]) Доказательство этого факта можно найти, например, в работе [6] на стр. 74.

Пример 8.7. Если $A = 11$, то 2 является примитивным корнем $GF(11)$. Из теоремы 8.5 следует, что $M(11, 3) = (2^{(11-1)/2} + 1)/11 = 3$. Следовательно, $11N$-код с $N_0 = 2$ имеет минимальное расстояние $d_{\text{мин}} = 3$ и может исправлять ошибки веса 1. Этот код имеет три кодовых числа $11 \cdot 0 = 0$, $11 \cdot 1 = 11$ и $11 \cdot 2 = 22$, которым соответствуют следующие кодовые слова: 00000, 01011 и 10110.

При этом ненулевыми вычетами по модулю $A = 11$ являются следующие числа:

2^i	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9
$2^i \bmod 11$	1	2	4	8	5	$10 \equiv -1$	$9 \equiv -2$	$7 \equiv -4$	$3 \equiv -8$	$6 \equiv -5$

Действительно, $\pm 2^i \not\equiv \pm 2^j \pmod{A}$ для всех i и j $(i \neq j)$, таких, что $|2^i + 2^j| < AM = 11 \cdot 3 = 33$. Каждому из этих вычетов соответствует одна из ошибок $\pm 2^0$; $\pm 2^1$; $\pm 2^2$; $\pm 2^3$; $\pm 2^4$ или правильное кодовое слово. Для справок ниже приведена таблица, в которой сопоставляются указанным выше 11 возможным ошибкам ($E = 0$ указывает на отсутствие ошибок) вычеты по модулю 11.

E	-2^4	-2^3	-2^2	-2^1	-2^0	0	2^0	2^1	2^2	2^3	2^4
$E \bmod 11$	6	3	7	9	10	0	1	2	4	8	5

В заключение обратим внимание на аналогию AN-кодов и обычных циклических кодов.

Пусть $A(X)$ — многочлен степени r над $GF(2)$, и пусть n — минимальное целое число, такое, что $A(X)$ делит $X^n - 1$. Тогда циклический код j-го порядка, порождаемый $A(X)$, будет представлять собой совокупность последовательностей $(f_0, f_1, f_2, \ldots, f_{jn-1})$, $f_i \in GF(2)$, $1 \leqslant i \leqslant jn$, длины jn, таких, что многочлен

$$f(X) = f_0 + f_1 X + f_2 X^2 + \ldots + f_{jn-1} X^{jn-1}$$

делится на $A(X)$. Следовательно, все многочлены $f(X)$, соответствующие кодовым словам, имеют вид

$$f(X) = A(X) N(X),$$

где $N(X)$ — многочлен степени $jn - r - 1$ или менее.

Если $A(X)$ имеет четное число ненулевых членов, то $A(1) = 0$. В действительности верно и обратное. В этом случае для всех

кодовых слов $A(1) = 0$, т. е. все кодовые слова имеют четный вес. Такие коды имеют минимальное расстояние Хэмминга, не меньшее 2, и, следовательно, могут обнаруживать ошибки веса 1 (в смысле расстояния Хэмминга). Многочлен $A(X) = 1 + X$ порождает код этого типа, имеющий минимальную избыточность, равную 1 при любом j. Такие коды полностью соответствуют AN-кодам, описанным в теореме 8.3.

Если многочлен $A(X)$ является примитивным, то $n = 2^r - 1$, и порождаемые $A(X)$ циклические коды первого порядка имеют минимальное расстояние 3 и являются кодами Хэмминга, исправляющими одиночные ошибки. Эти коды полностью соответствуют AN-кодам, даваемым теоремой 8.5. Кроме того, как было замечено сразу же после доказательства этой теоремы, и те и другие коды являются совершенными.

8.6. AN-коды, исправляющие кратные ошибки

В предыдущем разделе были описаны коды, исправляющие одиночные ошибки. Как показывают теоремы 8.4 и 8.5, такие коды могут строиться систематическим образом. Что касается AN-кодов, исправляющих кратные ошибки, то, за исключением специального случая описанных ниже циклических AN-кодов, общих методов их построения неизвестно.

В табл. 8.9 приведено несколько примеров AN-кодов с минимальным расстоянием $d_{\text{мин}} = 3$, 4 и 5. Эти коды были построены с помощью утверждения 8.2.1, а именно путем последовательного вычисления весов кодовых чисел $A \cdot 0$, $A \cdot 1$, $A \cdot 2$, ..., $A \cdot N$ с целью нахождения минимального числа $M(A, d)$, такого, что $A \cdot M(A, d) \leqslant d$. Однако объем вычислений резко возрастает с ростом A и d.

В настоящее время одним из полезных методов построения AN-кодов, исправляющих кратные ошибки, является построение по известным кодам с некоторым минимальным расстоянием d кодов с тем же минимальным расстоянием d, имеющих бóльшую длину. Здесь приведем следующую теорему, принадлежащую И. Л. Ерошу и С. Л. Ерошу [10].

Теорема 8.7. *Если A_1 порождает AN-код длиной n_1 с минимальным расстоянием d_1 и A_2 порождает AN-код длиной $n_2 (n_2 \geqslant n_1)$ с минимальным расстоянием $d_2 = [(d+1)/2]$, то при выполнении любого из следующих двух условий*:

$$\left.\begin{aligned} &2^{n_1} \equiv 1 \pmod{A_1}, \\ &(2^{n_1} - 1,\ A_2) = 1 \end{aligned}\right\} \tag{8.66}$$

и

$$\left.\begin{aligned} 2^{n_1} &\equiv -1 \pmod{A_1}, \\ (2^{n_1}+1,\ A_2) &= 1 \end{aligned}\right\} \tag{8.67}$$

число $A = A_1A_2$ *порождает* AN*-код длиной* $n (n \geqslant 2n_1)$ *с минимальным расстоянием* d_1.

Доказательство. Докажем теорему в предположении, что выполняется условие (8.66). При выполнении условия (8.67) доказательство проводится аналогично.

Предположим, что по крайней мере для одной совокупности чисел α_i или r_i, $\alpha_i \in \{0, +1, -1\}$, $0 \leqslant r_i \leqslant 2n_1 - 1$, имеет место сравнение

$$\sum_{i=1}^{d_1-1} \alpha_i 2^{r_i} \equiv 0 \pmod{A_1A_2}. \tag{8.68}$$

Пусть $r_i = n_1x_i + \rho_i$, где $x_i = 0$, если $0 \leqslant r_i \leqslant n_1 - 1$, и $x_i = 1$, если $n_1 \leqslant r_i \leqslant 2n_1 - 1$, а $\rho_i \in \{0, 1, \ldots, (n_1 - 1)\}$. Тогда формулу (8.68) можно переписать в виде

$$\sum_{i=1}^{d_1-1} \alpha_i 2^{n_1 + x_\lambda + \rho_i} \equiv 0 \pmod{A_1A_2}. \tag{8.69}$$

Эта система сравнений должна иметь место также по модулю A_1 и по модулю A_2. Поэтому

$$\sum_{i=1}^{d_1-1} \alpha_i 2^{\rho_i} \equiv 0 \pmod{A_1}. \tag{8.70}$$

(Здесь мы воспользовались также условием (8.66).) Так как A_1 порождает AN-код длиной n_1 с минимальным расстоянием d_1, то последнее сравнение может иметь место только в том случае, если

$$\sum_{i=1}^{d_1-1} \alpha_i 2^{\rho_i} = 0. \tag{8.71}$$

Кроме того, имеется еще система сравнений по мудулю A_2:

$$\sum_{i=1}^{d_1-1} \alpha_i 2^{n_i x_i + \rho_i} \equiv 0 \pmod{A_2}. \tag{8.72}$$

Если в формуле (8.72) $x_i = 0$ для d' слагаемых и $x_i = 1$ для d'' слагаемых $(d' + d'' = d_1 - 1)$, то

$$\sum_{j=1}^{d'} \alpha_j 2^{\rho_j} + \sum_{l=1}^{d''} \alpha_l 2^{n_1 + \rho_l} \equiv 0 \pmod{A_2}. \tag{8.73}$$

Если предположить здесь, что $d' \leqslant d''$, то $d' \leqslant [(d_1 - 1)/2]$. Прибавляя к обеим частям (8.73) $\sum_{j=1}^{d'} \alpha_j 2^{n_1 + \rho_j}$ и принимая во внимание равенство (8.71), получаем

$$(2^{n_1} - 1) \sum_{j=1}^{d'} \alpha_j 2^{\rho_j} \equiv 0 \pmod{A_2}. \tag{8.74}$$

Так как, согласно формулам (8.66), числа $2^{n_1} - 1$ и A_2 взаимно просты, то обе части последнего сравнения можно разделить на $2^{n_1} - 1$. В результате имеем

$$\sum_{j=1}^{d'} \alpha_j 2^{\rho_j} \equiv 0 \pmod{A_2}, \tag{8.75}$$

где $0 \leqslant \rho_i \leqslant n_1 - 1$ и $\alpha_j \in \{0, +1, -1\}$. Однако это сравнение не может выполняться, так как по условию теоремы A_2 порождает AN-код длиной $n_2 (n_2 \geqslant n_1)$ с минимальным расстоянием $d_2 = [(d_1 + 1)/2]$. Следовательно, не выполняется и система сравнений (8.68).

В случае если $d'' < d'$, прибавляя к обеим частям (8.73) число $\sum_{l=1}^{d''} \alpha_l 2^{\rho_l}$, приходим к тому же выводу.

Теорема 8.8. *Пусть A_1, A_2, A_3 — целые числа. Если целое число A_1 порождает AN-код длиной n_1 с минимальным расстоянием $d_1 = 5$, A_2 порождает AN-код длиной n_2 $(n_2 \geqslant n_1)$ с минимальным расстоянием $d_2 = 3$ и A_2A_3 порождает AN-код длиной $n_{2,3}$ $(n_{2,3} \geqslant 2n_1)$ с минимальным расстоянием $d_{2,3} = 3$, то при выполнении любого из следующих двух условий:*

$$\left.\begin{aligned} &2^{n_1} \equiv 1 \pmod{A_1}, \\ &(2^{n_1} \pm 1,\ A_2) = 1, \\ &2^{n_1} \equiv -1 \pmod{A_3} \end{aligned}\right\} \tag{8.76}$$

и

$$\left.\begin{aligned} &2^{n_1} \equiv -1 \pmod{A_1}, \\ &(2^{n_1} \pm 1,\ A_2) = 1, \\ &2^{n_1} \equiv 1 \pmod{A_3} \end{aligned}\right\} \tag{8.77}$$

число $A = A_1A_2A_3$ порождает AN-код длиной n $(n \geqslant 3n_1)$ с минимальным расстоянием $d = 5$.

Теорема 8.9. *Пусть A_1, A_2, A_3 и A_4 — целые числа. Если A_1 и A_2 порождают AN-коды длиной n_0 с минимальным расстоянием $d = 5$, такие, что $2^{n_0} \equiv 1 \pmod{A_1}$ и $2^{n_0} \equiv -1 \pmod{A_2}$, A_3 по-*

рождает AN-код длиной $n_3 \geqslant 4n_0$ *с минимальным расстоянием* $d_3 = 3$, A_3 *взаимно просто с* $2^{2n_0} - 1$ *и* $2^{2n_0} + 1$ *и* A_4 *является делителем* $2^{2n_0} + 1$, *то* $A = A_1A_2A_3A_4$ *порождает AN-код длиной n* ($n \geqslant 6n_0$) *с минимальным расстоянием* $d = 5$.

Теоремы 8.8 и 8.9 могут быть доказаны точно так же, как и теорема 8.7, но здесь эти доказательства опускаются [1]). Приведем некоторые утверждения, полезные при построении кодов с помощью этих теорем.

Сначала заметим, что любое целое число A вида

$$A = \frac{2^{kd} + 1}{2^k + 1}, \quad d \text{ — нечетное число,} \tag{8.78}$$

или

$$A = \frac{2^{kd} - 1}{2^k + 1}, \quad d \text{ — четное число,} \tag{8.79}$$

где $k = 2, 3, 4, \ldots$, порождает AN-код длиной $n = kd$ с минимальным расстоянием d. Действительно, легко доказать, что для любого целого числа A указанного выше вида $M(A, d) = 2^k + 1$ и $W(AM(A, d)) = 2$.

Далее, при любом целом положительном d число

$$A = (2^{kd} - 1)/(2^k - 1) \tag{8.80}$$

порождает AN-код длины $n = kd$ с минимальным расстоянием d. При этом $M(A, d) = 2^k - 1$ и $W(AM(A, d)) = 2$.

Минимальным целым положительным числом, имеющим вес d, является

$$A = (2^{2d-1} + 1)/3, \tag{8.81}$$

но для таких A $M(A, d) = 3$.

В табл. 8.10 приведено несколько кодов с минимальным расстоянием 5, построенных с помощью теорем 8.7—8.9.

8.7. Синдромы и методы декодирования *AN*-кодов

Предположим, что кодовое число AN в результате арифметической ошибки E переходит в число $I = AN + E$. Минимальный вычет по модулю A числа I:

$$r_A(I) = r(AN + E) = r_A(E) \tag{8.82}$$

определяется только ошибкой E. Этот вычет будем называть синдромом числа I. Естественно, что при отсутствии ошибок, т. е. при $E = 0$, $r_A(I) = 0$.

---

[1]) См. [10].

Таблица 8.10

Примеры *AN*-кодов с минимальным расстоянием 5

A	A_1	A_2	A_3	A_4	Длина кода n
27 531	171	23	7		27
39 215	341	23	5		30
14 145	205	23	3		30
62 715	565	37	3		42
98 635 955	205	341	83	17	60
463 185	1 971	47	5		54
199 797	1 417	47	3		54
11 163 506 255	3 641	4 681	131	5	90
7 739 145 297	1 417	1 971	163	17	108

Связь между свойствами синдрома и корректирующей способностью кода устанавливается следующей теоремой, являющейся обобщением леммы 8.3.

Теорема 8.10. *Пусть* ξ *— некоторое множество ошибок. Для того чтобы AN-код исправлял все ошибки из множества* ξ*, необходимо и достаточно, чтобы любые две ошибки* E_i *и* E_j*, такие, что* $E_i \neq E_j$ *и*

$$|E_i - E_j| \leqslant AN_0, \tag{8.83}$$

имели различные синдромы.

Доказательство. Пусть E_1 и E_2 — две различные ошибки. Если допустить, что эти ошибки нельзя исправить, то найдутся такие кодовые числа AN_1 и AN_2, что

$$AN_1 + E_1 = AN_2 + E_2.$$

Так как $0 \leqslant N_1 \leqslant N_0$ и $0 \leqslant N_2 \leqslant N_0$, то $|E_1 - E_2| = |A(N_2 - N_1)| \leqslant AN_0$. Однако $r_A(E_1) = r_A(AN_1 + E_1) = r_A(AH_2 + E_2) = r_A(E_2)$.

Следовательно, код исправляет все ошибки из ξ тогда и только тогда, когда $r_A(E_1) \neq r_A(E_2)$ для любых двух различных ошибок E_1 и E_2, удовлетворяющих неравенству (8.83)[1]).

Казалось бы, что в данной формулировке этой теоремы условие (8.83) можно исключить, а необходимое и достаточное условие исправления кодом ошибок заданного множества сформулировать следующим образом: «все ошибки из ξ имеют попарно различные синдромы». В действительности, как следует из тео-

[1]) Необходимость условия (8.83) не доказана. — *Прим. ред.*

ремы 8.10, это не так; последнее условие является достаточным, но не необходимым.

Рассмотрим AN-код с $A = 11$ и $N_0 = 2$ из примера 8.7. Пусть ξ — множество, состоящее из ошибок E веса 1, таких, что $|E| < A \cdot M\ (11, 3) = 33$, и числа 0. Этот AN-код имеет минимальное расстояние 3 и, следовательно, может исправлять все ошибки веса 1. Ошибки из множества ξ удовлетворяют условиям теоремы 8.10, но не обязательно все они имеют различные синдромы. Например, ошибки $E_1 = 2^5$ и $E_2 = -1$ входят в ξ, но $r_A(E_1) = r_A(E_2) = 10$. Однако поскольку $|E_1 - E_2| = 33 > AN_0 = 22$, то условия теоремы 8.10 в действительности и не требуют, чтобы ошибки имели различные синдромы. Тем не менее обе эти ошибки рассматриваемым AN-кодом исправляются. Это происходит потому, что ни для каких двух кодовых чисел AN_1 и AN_2 из трех кодовых чисел 0, 11 и 22 не выполняется равенство $AN_1 + E_1 = AN_2 + E_2$. Рассмотрим, например, уравнение $I = 11 + E_1 = 11 + 32 = 43$. В этом случае $r_A(I) = r_A(11 + E_1) = r_A(43) = 10$ и, следовательно, возникла одна из двух ошибок 2^5 или -1. Однако число $43 - (-1) = 44 > 33$ не является кодовым числом рассматриваемого кода, и, значит, ошибка -1 возникнуть не могла. Допустим, что произошла ошибка 2^5. Тогда $43 - 2^5 = 43 - 32 = 11$. В результате приходим к выводу, что ошибка равна 2^5, а правильное кодовое число равно 11.

Таким образом, исправляемые ошибки не обязательно должны иметь различные синдромы. Кроме того, непосредственным следствием теоремы 8.10 является следующее утверждение.

Утверждение 8.10.1. *AN-код исправляет все ошибки из заданного множества ξ тогда и только тогда, когда при любом фиксированном целом i все ошибки из ξ, удовлетворяющие условию*

$$i \leqslant E \leqslant i + AN_0, \tag{8.84}$$

имеют различные синдромы.

Это утверждение непосредственно следует из теоремы 8.10, если заметить, что когда E_l и $E_{l'}$ удовлетворяют неравенству (8.84) при одном и том же i, то они удовлетворяют также и (8.83).

Последнее утверждение оказывается полезным, когда на величину ошибок устанавливается ограничение. Так как число $I = AN + E$ лежит в области $0 \leqslant I \leqslant 2^n - 1$, где n — длина кода, задаваемая равенством (8.32), то

$$-AN \leqslant E \leqslant 2^n - 1 - AN.$$

Однако поскольку $0 \leqslant N \leqslant N_0$, то это неравенство можно заменить на следующее неравенство, не связанное с N:

$$-AN_0 \leqslant E \leqslant 2^n - 1. \tag{8.85}$$

Ошибки E, удовлетворяющие (8.85), называются реально возможными ошибками. Для рассмотренного выше AN-кода с $A = 11$ и $N_0 = 2$ реально возможными ошибками являются ошибки, удовлетворяющие неравенству

$$-22 \leqslant E \leqslant 31.$$

Как видно из таблицы, приведенной в примере 8.7, целые числа веса 0 и 1, лежащие в этой области (это множество обозначается через ξ), имеют различные синдромы. Действительно, для AN-кодов, даваемых теоремой 8.5 из разд. 8.5, реально возможные ошибки веса 0 и 1 имеют различные синдромы, и, более того, эти синдромы «пробегают» все вычеты по модулю A.

Однако, вообще говоря, различные синдромы могут иметь не только исправляемые реально возможные ошибки. Например, в приведенном выше примере $E_1 = 17$ является исправляемой ошибкой веса 2. Если эту ошибку ввести в ξ, то все элементы ξ будут исправляемыми ошибками и ξ по-прежнему останется множеством реально возможных ошибок. При этом синдром $r_A(E_1) = r_A(17) = 6$ равен синдрому $r_A(-16) = 6$ ошибки $E_2 = -16$. Однако поскольку $|E_1 - E_2| = 33 > AN_0 = 22$, то теорема 8.10 и не требует, чтобы ошибки E_1 и E_2 имели различные синдромы.

Остановимся здесь кратко на декодировании AN-кодов. Простейший способ состоит в том, чтобы построить таблицу декодирования, в которой каждому из 2^n возможных чисел $I = AN + E$ сопоставлялось бы правильное кодовое число. Однако при достаточно больших n для хранения этой таблицы требуется память большой емкости, а кроме того, большим оказывается и время обращения к этой таблице. Декодирование значительно упрощается при использовании синдрома. Вычислив синдром $r_A(I) = r_A(E)$ и найдя по таблице ошибку E, соответствующую этому синдрому, можно восстановить правильное кодовое число AN, вычитая E из I. Так как всего имеется не более A различных вычетов, то в этом случае используемая таблица имеет не более A разделов. Если синдрому $r_A(I)$ соответствуют две или более ошибки, то из этих ошибок необходимо выбрать ту ошибку, после вычитания которой из I получается кодовое число (такая ошибка единственна).

Как можно видеть из этого краткого пояснения, желательно, чтобы соответствие между реально возможными исправляемыми ошибками и соответствующими им синдромами было взаимно однозначным. Такие коды известны [11], но даже если это условие не выполняется, можно использовать другие описанные ниже методы декодирования. Кроме того, в случае, если длина кода такова, что $2^n - 1 \equiv 0 \pmod{A}$, полезным оказывается алгоритм декодирования, описанный в разд. 9.3.

9. Циклические AN-коды

9.1. Структура циклических AN-кодов[1])

9.1.1. Определение циклического AN-кода

Если множество кодовых слов замкнуто относительно *циклических сдвигов*, т. е. если циклические сдвиги любого кодового слова вновь приводят к кодовым словам, то такой AN-код называется *циклическим* AN-кодом. В частности, если любое кодовое слово AN-кода может быть получено циклическими сдвигами из любого другого кодового слова, то такой код называется *сильно циклическим*.

Например, рассмотрим AN-код с $A = 9$ и $N_0 = 6$. Кодовыми числами этого кода являются 0, 9, 18, 27, 36, 45, 54, которым соответствуют кодовые слова 000000, 001001, 010010, 011011, 100100, 101101, 110110. Это множество кодовых слов является замкнутым относительно циклических сдвигов, и, следовательно, рассматриваемый AN-код является циклическим.

Циклические AN-коды впервые исследовал Мандельбаум [1]. После этого они были предметом большого числа публикаций других авторов. В этом разделе рассматриваются основные свойства этих кодов.

9.1.2. Кольцо целых чисел по модулю $2^e - 1$ и его идеалы

Если целые числа 2 и B взаимно просты, т. е. $(2, B) = 1$ и показатель 2 по модулю B равен e, то, как следует из изложенного в разд. 8.1.7,

$$2^e \equiv 1 \pmod{B}. \tag{9.1}$$

Следовательно, при подходящем выборе целого числа A

$$2^e - 1 = A \cdot B. \tag{9.2}$$

Полная система вычетов по любому модулю является коммутативным кольцом (см. разд. 8.1.5). Пусть Re — кольцо целых чисел по модулю $2^e - 1$. Рассмотрим подмножество кольца Re, состоящее из чисел, кратных A:

$$\{0,\ A \cdot 1,\ A \cdot 2,\ \ldots,\ A \cdot (B - 1)\}.$$

---

[1]) Все изложенное в этой главе справедливо для системы счисления с произвольным основанием r, используемой для представления кодовых чисел, но, как и в предыдущей главе, ограничимся лишь случаем двоичной системы счисления, когда $r = 2$.

Используя равенство (9.2), легко показать, что это подмножество, состоящее из B целых чисел и включающее 0, является идеалом. Обозначим этот идеал через I, а число A назовем его порождающим элементом [1]).

Каждый элемент идеала I является числом, которое делится на A, и, следовательно, идеал I является AN-кодом. Следующая теорема показывает, что этот AN-код является циклическим.

Теорема 9.1. *Циклический сдвиг двоичного представления произвольного целого числа идеала I, т. е. кодового слова, также принадлежит идеалу I [2]). Другими словами, идеал I замкнут относительно циклических сдвигов.*

Доказательство. Так как $2^e \equiv 1 \pmod{(2^e - 1)}$, то двоичное представление произвольного числа N, входящего в I, имеет не более e разрядов. Пусть

$$N = \sum_{i=0}^{e-1} a_i 2^i.$$

Вновь пользуясь тем, что $2^e \equiv 1$, получаем

$$2N = \sum_{i=0}^{e-1} a_i 2^{i+1} = \sum_{i=0}^{e-1} a_i 2^{i+1} + a_{e-1} 2^e \equiv a_{e-1} + \sum_{i=1}^{e-1} a_{i-1} 2^i \pmod{2^e - 1}.$$

Таким образом, двоичное представление $2N \pmod{(2^e - 1)}$ является циклическим сдвигом двоичного представления N. Так как 2 является элементом Re и N принадлежит I, то $2N \pmod{2^e - 1}$ также принадлежит I. Следовательно, двоичное представление $2N \pmod{2^e - 1}$ является кодовым словом.

Введенный таким образом циклический AN-код является идеалом кольца целых чисел по модулю $2^e - 1$, и все его элементы являются кратными порождающего элемента A. Сумма по модулю $2^e - 1$ произвольных двух кодовых слов идеала также принадлежит идеалу и, следовательно, является кодовым словом. Кроме того, как видно из вышеизложенного, e определяет длину кода, а B — число кодовых слов.

---

[1]) Подмножество S кольца R называется идеалом кольца R, если оно удовлетворяет следующим трем условиям: 1) S замкнуто относительно сложения; 2) для произвольного элемента из S в S существует обратный элемент по сложению; 3) произведение произвольного элемента из S и произвольного элемента из R принадлежит S. Идеалы кольца Re являются главными идеалами. Детальное изложение этих вопросов можно найти, например, в [3].

[2]) Ниже кодовые числа часто называются также кодовыми словами. В этом случае имеется в виду двоичное представление этого числа

9.1.3. Задание циклических кодов

Из предыдущего раздела следует, что все кодовые слова циклического AN-кода длины $e = E(2, B)$, содержащего B кодовых слов, можно построить по двоичному представлению порождающего числа A, которое определяется из равенства $2^e - 1 = AB$. Произвольное кодовое слово может быть построено путем проведения циклических сдвигов двоичного представления A и вычисления определенных линейных комбинаций кодовых слов, получающихся в результате указанных циклических сдвигов. В этом разделе, представив порождающее число A в виде

$$A = \sum_{j=0}^{e-1} a_j 2^j, \quad a_j \in \{0, 1\},$$

докажем следующие сравнения [1]):

$$a_{e-i}B \equiv -2^i \bmod B \pmod 2, \quad 1 \leqslant i \leqslant e,$$

а кроме того, укажем простой способ определения коэффициентов a_i, $0 \leqslant i \leqslant e-1$, двоичного представления числа A.

Обращаясь к разд. 8.1.11 и равенству (9.2), имеем

$$\frac{1}{B} = \frac{A}{2^e - 1} = A(0, \dot{0}\underbrace{0 \ldots 0}_{\text{период } e}\dot{1}). \tag{9.3}$$

Так как $A < 2^e - 1$, то длина двоичного представления A не превосходит e. Полагая

$$A = (a_{e-1}a_{e-2} \ldots a_2a_1a_0),$$

из формулы (9.3) получаем

$$\frac{1}{B} = 0, \dot{a}_{e-1}a_{e-2} \ldots a_2a_1\dot{a}_0.$$

Следовательно, при любом целом i, $1 \leqslant i \leqslant e$, имеет место следующее равенство:

$$\frac{2^i}{B} = a_{e-1}a_{e-2} \ldots a_{e-i}, \quad a_{e-i-1} \ldots a_0\dot{a}_{e-1} \ldots \dot{a}_0 = q + \frac{2^i \bmod B}{B},$$

где q и $2^i \bmod B/B$ являются соответственно целой и дробной частями числа $2^i/B$. Отсюда в свою очередь следует, что

$$q = a_{e-1}a_{e-2} \ldots a_{e-i} = \frac{2^i}{B} - \frac{2^i \bmod B}{B}$$

и

$$(a_{e-1}a_{e-2} \ldots a_{e-i})B = 2^i - 2^i \bmod B. \tag{9.4}$$

[1]) Напомним, что под $2^i \bmod B$ понимается наименьший неотрицательный вычет 2^i по модулю B.

Переходя к сравнению по модулю 2, получаем

$$a_{e-i}B \equiv -2^i \bmod B \pmod 2 \tag{9.5}$$

или (поскольку $2 \bmod 2 = 0$)

$$a_{e-i}B \equiv (2 - 2^i \bmod B) \bmod 2. \tag{9.6}$$

Так как $(2, B) = 1$, то $B \bmod 2 = 1$ и, следовательно, $a_{e-i}B \bmod 2 = a_{e-i} \bmod 2 \cdot B \bmod 2 = a_{e-i} \bmod 2$. Так как при этом $a_{e-i} \in \{0, 1\}$, то $a_{e-i} \bmod 2 = a_{e-i}$. Кроме того, $(2 - 2^i \bmod B) \cdot \bmod 2 = (2^i \bmod B) \bmod 2$. Таким образом,

$$a_{e-i} = (2^i \bmod B) \bmod 2, \tag{9.7}$$

т. е. $a_{e-i} = 1$, если $2^i \bmod B$ — нечетное число, и $a_{e-i} = 0$, если $2^i \bmod B$ — четное число.

Например, при $B = 13$, $2^{12} - 1 \equiv 0 \pmod{13}$ получаем двоичное представление A, приведенное в табл. 9.1.

Таблица 9.1

Двоичное представление A при $B = 13$

i	1	2	3	4	5	6	7	8	9	10	11	12
$2^i \bmod B$	2	4	8	3	6	12	11	9	5	10	7	1
A	0	0	0	1	0	0	1	1	1	0	1	1

Действительно, 12-разрядным двоичным представлением $A = (2^{12} - 1)/B = 4095/13 = 315$ является 000100111011. Путем деления двоичных чисел можно проверить, что $1/B = 1/13 = 0{,}\dot{0}00100111011\dot{1}$.

Произвольное ненулевое кодовое слово этого AN-кода может быть получено путем циклического сдвига $A \cdot 1$.

9.1.4. Разложение и структура циклических AN-кодов

Циклические AN-коды разлагаются на подкоды, каждый из которых в свою очередь может быть разложен на несколько простых кодов. Простые коды являются строго циклическими, и, следовательно, путем циклических сдвигов их кодовых слов можно построить исходный код. Это широко используется при определении минимального расстояния AN-кодов. В данном разделе на основе разложения мультипликативной группы, описанного в разд. 8.1.8, будет проведено разложение циклического AN-кода и таким образом описана структура последнего.

В результате циклических сдвигов кодового числа $A \cdot 1 = (a_{e-1}, a_{e-2}, \ldots, a_1, a_0)$ получаются кодовые слова

$A2^j \bmod (2^e - 1)$, $0 \leqslant j \leqslant e - 1$. Пользуясь тем, что $kAB = k(2^e - 1) \equiv 0 \pmod{(2^e - 1)}$ при любом целом k, все эти кодовые слова можно представить в виде

$$A2^j = A(2^j \bmod B + kB) \equiv A(2^j \bmod B) \pmod{2^e - 1}. \quad (9.8)$$

Попытаемся найти аналогичное сравнение в общем случае, т. е. для кодовых слов, получающихся в результате циклических сдвигов кодового слова $A\alpha$ $(0 < \alpha < B)$. Пусть $(b_{e-1}, b_{e-2}, \ldots, b_1, b_0)$ — двоичное представление $A\alpha$. Следуя выводу формулы (9.5) (соответствующему случаю $A \cdot 1$), можно проверить, что

$$b_{e-i}B \equiv -\alpha 2^i \bmod B \pmod 2. \quad (9.9)$$

Далее представим кодовые слова, получающиеся в результате циклических сдвигов $A\alpha$, в виде

$$A\alpha 2^j \equiv A(\alpha 2^j \bmod B) \pmod{(2^e - 1)}. \quad (9.10)$$

Следовательно, если α выбрать так, что $\alpha \leqslant \alpha 2^j \bmod B$, то всем ненулевым кодовым словам будут соответствовать ненулевые вычеты по модулю B.

Далее, как описывалось в разд. 8.1.8, приведенная система вычетов по модулю B, т. е. множество вычетов, взаимно простых с B, образует мультипликативную группу $G(B)$, и если $G(B)$ имеет хотя бы одну нетривиальную подгруппу, то $G(B)$ можно разложить на непересекающиеся классы. Так как между вычетами по модулю B и кодовыми словами существует описанное выше соответствие, то циклические AN-коды также могут быть разбиты на подкоды, как описывается ниже. Множество кодовых слов, соответствующих вычетам, входящим в $G(B)$, назовем *подкодом* и обозначим через $S(B)$. Путем разложения $G(B)$ на смежные классы по подгруппе можно разложить $S(B)$ на *простые коды*. $S(B)$ может разлагаться на несколько простых кодов, каждый из которых будет сильно циклическим и состоять из кодовых слов, получающихся циклическим сдвигом некоторого $A\alpha$, $\alpha \in G(B)$. Этот простой код будем обозначать через $C_\alpha(B)$.

Очевидно, что подкод $S(B)$ состоит из $\varphi(B)$ кодовых слов. Так как $e = E(2, B)$ по определению является минимальным целым числом k, таким, что $\alpha \equiv \alpha 2^k \pmod B$, то каждый простой код содержит e кодовых слов. Следовательно, $S(B)$ состоит из $\nu = \varphi(B)/e$ простых подкодов.

Если B является простым числом, то $G(B)$ содержит все вычеты по модулю B, и, следовательно, $S(B)$ содержит все кодовые слова и других подкодов, кроме $S(B)$, не существует. Далее, если при этом число 2 является примитивным корнем B, то $e = \varphi(B)$ и, следовательно, существует один простой код, кото-

рый к тому же совпадает с $S(B)$. Таким образом, если B — простое число и 2 — примитивный корень B, то может быть построен строго циклический AN-код.

Однако если B не является простым, то существует вычет α по модулю B, не являющийся взаимно простым с B. Предположим, что $(\alpha, B) = d$. Тогда, поскольку $(\alpha/d,\ B/d) = 1$, то $\alpha/d \in G(B/d)$. В этом случае множество кодовых слов, соответствующих вычетам по модулю B, получающимся умножением на d вычетов по модулю B/d, взаимно простых с B/d, образует подкод. Обозначим этот подкод через $S(B)_d$. В $S(B)_d$ входит $\varphi(B/d)$ кодовых слов. Если показатель 2 по модулю B/d равен k, а именно если $2^k \equiv 1 \pmod{B/d}$, то каждый простой код в $S(B)_d$ содержит k кодовых слов. Само собой разумеется, что k является делителем e. Следовательно, $S(B)_d$ состоит из $\nu_d = \varphi(B/d)/k$ простых кодов. Здесь следует заметить, что $S(B)_d$ равно (e/k)-кратному повторению подкода $S(B/d)$, соответствующего вычетам из $G(B/d)$.

Таким образом, циклический AN-код раскладывается на подкоды $S(B)_{d_i}$, соответствующие различным делителям d_i числа B и содержащие $\varphi(B/d_i)$ кодовых слов. При этом в каждом из кодов $S(B)_{d_i}$ существует $\nu_{d_i} = \varphi(B/d_i)/k_i$ строго циклических простых кодов, каждый из которых содержит $k_i = E(2, B/d_i)$ кодовых слов. Подкод $S(B)_{d_i}$ содержит ровно $\nu = \nu_{d_i}$ простых кодов:

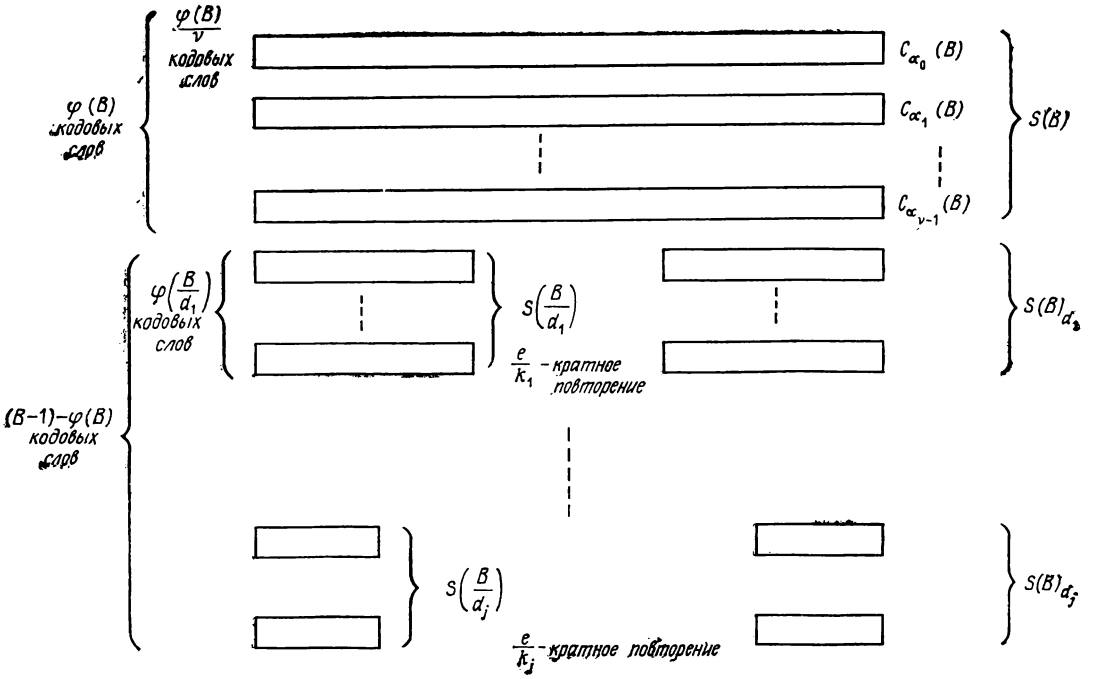
$$S(B)_{d_i} = \{C_{\alpha_1}(B),\ C_{\alpha_2}(B),\ \ldots,\ C_{\alpha_k}(B),\ \ldots,\ C_{\alpha_\nu}(B)\},$$

где α_k $\left(k = 1, \ldots, \nu = \nu_{d_i}\right)$ — целые числа, удовлетворяющие условию $(\alpha_k, B) = d_i$. Схема разложения циклического AN-кода на подкоды, а подкодов на простые коды показана на фиг. 9.1, которая позволяет также понять структуру циклического AN-кода.

Если разложению подкода на простые коды соответствует разложение мультипликативной группы на смежные классы по ее подгруппе, то разложение циклического AN-кода на подкоды является одним из применений равенства (8.12):

$$\sum_{1 \leqslant d \leqslant B} \varphi\left(\frac{B}{d}\right) = B.$$

Это равенство дает суммарное число кодовых слов B (включая кодовое слово 0, соответствующее случаю $d = B$). Разложение на подкоды соответствует разложению целого числа на слагаемые. Остановимся на этом несколько более подробно.



Фиг. 9.1. Разложение циклического AN-кода на подкоды и простые коды.

В целях упрощения рассмотрим случай, когда B разлагается в произведение l различных простых чисел, а именно когда

$$B = p_1 \cdot p_2 \cdot \ldots \cdot p_l,$$

где p_i $(1 \leqslant i \leqslant l)$ — простые числа. От этого случая легко перейти к другим возможным ситуациям. Исходя из разложения B на простые сомножители, можно построить следующую каскадную схему подкодов. Начиная с $S(B)$, соответствующего случаю $d = 1$, рассмотрим $S(B)_{d=p_i}$, $S(B)_{d=p_ip_j}$, ..., $S(B)_{d=B}$. Подкоды можно классифицировать следующим образом: $S(B)$ — подкод 0-го порядка, $S(B)_{p_i}$ $(i = 1, \ldots, l)$ — подкоды 1-го порядка, $S(B)_{p_ip_j}$ $(i \neq j,\ i, j = 1, \ldots, l)$ — подкоды 2-го порядка, ..., $S(B)_B$ — подкод l-го порядка. Подкоды m-го порядка, $0 \leqslant m \leqslant l$, представляют собой коды $S(B)_d$, где d — произведение m сомножителей из разложения B; всего существует $\binom{l}{m}$ подкодов m-го порядка. В частности, подкод l-го порядка состоит только из одного нулевого кодового слова. Такая структура называется «пучковой» конструкцией. Пучковая конструкция подкодов в случае $B = p_1p_2p_3p_4$ показана на фиг. 9.2.

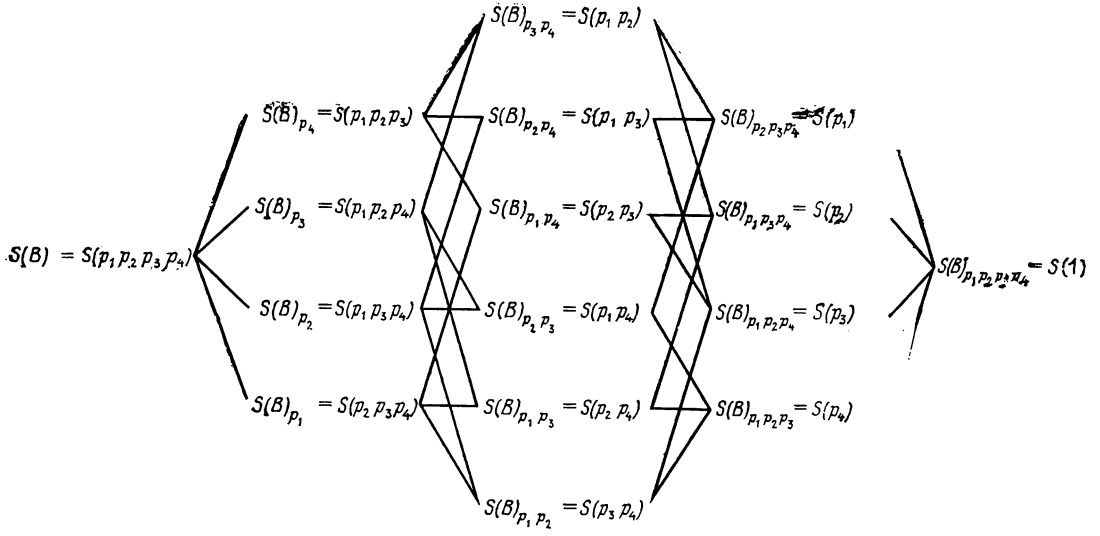
Приведем пример, показывающий, каким образом циклический AN-код разлагается на подкоды и простые коды.

Пример 9.1. Рассмотрим случай $B = 105$. При этом $e = E(2, 105) = 12$, $AB = (2^e - 1) = 2^{12} - 1 = (3 \times 13) \times (3 \times 5 \times 7)$, $A = 39$. Как следует из изложенного в разд. 9.1.2, рассматриваемый код имеет длину $e = 12$ и содержит 105 кодовых чисел. Поскольку B является составным числом, у кода существует несколько подкодов. Подкод 0-го порядка $S(B) = S(105)$ приведен в табл. 9.2. Кодовые слова, приведенные в этой таблице, были вычислены с помощью формулы (9.7) или (9.9). Каждый простой код является строго циклическим, и все его кодовые слова могут быть получены путем циклических сдвигов кодового слова, помещенного в соответствующей графе формулы. Каждый простой код содержит 12 кодовых слов; всего в $S(B)$ содержится $\varphi(B) = \varphi(3) \cdot \varphi(5) \cdot \varphi(7) = 48$ кодовых слов.

Существуют три подкода 1-го порядка: $S(B)_3$, $S(B)_5$, $S(B)_7$; эти подкоды приведены в табл. 9.3.

Существуют три подкода 2-го порядка: $S(B)_{15}$, $S(B)_{21}$, $S(B)_{35}$; эти подкоды приведены в табл. 9.4.

В подкоды 1-го и 2-го порядка входят кодовые слова, соответствующие ненулевым вычетам по модулю B, которые не являются взаимно простыми с B; всего имеется $(B - 1) - \varphi(B) = (105 - 1) - 48 = 56$ таких кодовых слов. Эти подкоды эквивалентны подкодам, соответствующим вычетам, которые при



Фиг. 9.2. Пучковая конструкция из подкодов.

Таблица 9.2

Подкод 0-го порядка $S(B)$

$S(B) = S(105) = \{C_1(B),\ C_{11}(B),\ C_{13}(B),\ C_{17}(B)\}$

	l	1	2	3	4	5	6	7	8	9	10	11	12
$C_1(B)$	$2^l \bmod 105$	2	4	8	16	32	64	23	46	92	79	53	1
	$A \cdot 1$	0	0	0	0	0	0	1	0	0	1	1	1
$C_{11}(B)$	$11 \cdot 2^l \bmod 105$	22	44	88	71	37	74	43	86	67	29	58	11
	$A \cdot 11$	0	0	0	1	1	0	1	0	1	1	0	1
$C_{13}(B)$	$13 \cdot 2^l \bmod 105$	26	52	104	103	101	97	89	73	41	82	59	13
	$A \cdot 13$	0	0	0	1	1	1	1	1	1	0	1	1
$C_{17}(B)$	$17 \cdot 2^l \bmod 105$	34	68	31	62	19	38	76	47	94	83	61	17
	$A \cdot 17$	0	0	1	0	1	0	0	1	0	1	1	1

Таблица 9.3

Подкоды 1-го порядка: $S(B)_3$, $S(B)_5$, $S(B)_7$

а) $S(B)_3 = \{C_3(B),\ C_9(B)\}$

$C_3(B)$	$3 \cdot 2^l \bmod 105$	6	12	24	48	96	87	69	33	66	27	54	3
	$A \cdot 3$	0	0	0	0	0	1	1	1	0	1	0	1
$C_9(B)$	$9 \cdot 2^l \bmod 105$	18	36	72	39	78	51	102	99	93	81	57	9
	$A \cdot 9$	0	0	0	1	0	1	0	1	1	1	1	1

б) $S(B)_5 = \{C_5(B),\ C_{25}(B)\}$

$C_5(B)$	$5 \cdot 2^l \bmod 105$	10	20	40	80	55	5	10	20	40	80	55	5
	$A \cdot 5$	0	0	0	0	1	1	0	0	0	0	1	1
$C_{25}(B)$	$25 \cdot 2^l \bmod 105$	50	100	95	85	65	25	50	100	95	85	65	25
	$A \cdot 25$	0	0	1	1	1	1	0	0	1	1	1	1

в) $S(B)_7 = \{C_7(B),\ C_{49}(B)\}$

$C_7(B)$	$7 \cdot 2^l \bmod 105$	14	28	56	7	14	28	56	7	14	28	56	7
	$A \cdot 7$	0	0	0	1	0	0	0	1	0	0	0	1
$C_{49}(B)$	$49 \cdot 2^l \bmod 105$	98	91	77	49	98	91	77	49	98	91	77	49
	$A \cdot 49$	0	1	1	1	0	1	1	1	0	1	1	1

Таблица 9.4

Подкоды 2-го порядка: $S(B)_{15}$, $S(B)_{21}$, $S(B)_{35}$

а) $S(B)_{15} = \{C_{15}(B), C_{45}(B)\}$

$C_{15}(B)$	$15 \cdot 2^j \bmod 105$ $A \cdot 15$	30 0	60 0	15 1	30 0	60 0	15 1	30 0	60 0	15 1	30 0	60 0	15 1
$C_{45}(B)$	$45 \cdot 2^j \bmod 105$ $A \cdot 45$	90 0	75 1	45 1	90 0	75 1	45 1	90 0	75 1	45 1	90 0	75 1	45 1

б) $S(B)_{21} = \{C_{21}(B)\}$

$C_{21}(B)$	$21 \cdot 2^j \bmod 15$ $A \cdot 21$	42 0	84 0	63 1	21 1	42 0	84 0	63 1	21 1	42 0	84 0	63 1	21 1

в) $S(B)_{35} = \{C_{35}(B)\}$

$C_{35}(B)$	$35 \cdot 2^j \bmod 105$ $A \cdot 35$	70 0	35 1	70 0	35 1	70 0	35 1	70 0	35 1	70 0	35 1	70 0	35 1

надлежат $G(B/d)$. Например, подкод $S(B)_3$ эквивалентен подкоду $S(B/3) = S(35)$, кодовые слова которого соответствуют вычетам, принадлежащим $G(35)$ $(B/d = 105/3 = 35)$. Как видно из табл. 9.5, подкод $S(35)$ состоит из двух простых подкодов $C_1(35)$ и $C_3(35)$.

Таблица 9.5

$S(35) = \{C_1(35), C_3(35)\}$

$C_1(35)$	$2^j \bmod 35$ $A' \cdot 1$	2 0	4 0	8 0	16 0	32 0	29 1	23 1	11 1	22 0	9 1	18 0	1 1
$C_3(35)$	$3 \cdot 2^j \bmod 35$ $A' \cdot 3$	6 0	12 0	24 0	13 1	26 0	17 1	34 0	33 1	31 1	27 1	19 1	3 1

Простые подкоды $C_3(105)$ и $C_9(105)$ совпадают соответственно с $C_1(35)$ и $C_3(35)$. Далее, поскольку $k = E(2, 35) = 12$, то, как известно, в $C_1(35)$ и $C_3(35)$ содержится по 12 подкодовых слов. Рассмотрим подкод $S(B)_{21}$, соответствующий случаю $d = 21$, $B/d = 5$, $k = E(2, 5) = 4$. Этот код порождается 4-кратным повторением подкода $S(5)$, приведенного в табл. 9.6. Так как 4-кратный циклический сдвиг приводит к тому же кодовому слову, то этому подкоду принадлежит всего $\varphi(5) = 4$ кодовых слова.

Таблица 9.6

Построение $S\ (105)_{21}$ повторением $S\ (5)$

$2^i \bmod 5$	2	4	3	1	2	4	3	1	2	4	3	1
$A^r \cdot 1$	0	0	1	1	0	0	1	1	0	0	1	1

9.2. Минимальное расстояние циклических *AN*-кодов

В предыдущем разделе был описан способ построения циклических AN-кодов. После того как здесь будут получены оценки минимального расстояния этих кодов, с помощью теоремы 8.1 можно будет оценить их способности исправлять и обнаруживать ошибки. Как следует из теоремы 8.2, минимальное расстояние AN-кода равно минимальному из весов кодовых слов, принадлежащих этому AN-коду. Кроме того, как мы уже знаем, простые коды являются строго циклическими, и, следовательно, грубо говоря, вес одного кодового слова простого кода определяет минимальный вес последнего [1]). Таким образом, взяв по одному кодовому слову из каждого простого кода, определив веса этих кодовых слов и выбрав среди них минимальный, можно узнать минимальное расстояние циклического AN-кода.

В соответствии с равенством (3.2) здесь будем считать, что

$$2^n - 1 = A \cdot B. \tag{9.11}$$

9.2.1. Минимальное расстояние простых кодов

Рассмотрим простой код $C_\alpha(B)$. Предположим, что кодовые слова этого кода представлены в каноническом виде, т. е. имеют представления, удовлетворяющие условию M [2]. Для пояснения возьмем код $C_{13}(B) = C_{13}(105)$ из примера 9.1:

$$A \cdot 13 = (000111111011) = (001000000\bar{1}0\bar{1}).$$

Следовательно, $W(A \cdot 13) = 3$. Очевидно, что кодовые слова, получающиеся в результате любых циклических сдвигов кодового числа $A \cdot 13$, представленного в каноническом виде, имеют одинаковый вес. В этом смысле все кодовые слова, принадлежащие $C_{13}(B)$, имеют один и тот же вес 3.

Следует однако, обратить внимание на одну трудность, связанную с тем, что среди кодовых слов, получающихся в результате циклических сдвигов канонического представления, имеют-

[1]) Этот вопрос требует детального исследования и будет подробно рассмотрен ниже.

[2]) См разд. 8.3.

ся слова, являющиеся представлением отрицательных чисел. Это связано с тем, что, хотя числа $A \cdot (\alpha \cdot 2^j) \bmod B$ и являются положительными, существуют отрицательные числа, сравнимые с ними по модулю $2^n - 1$. Поэтому, если выписать представления минимального веса для всех кодов, получающихся в результате циклических сдвигов обычного двоичного представления числа $A \cdot \alpha$, и определить их веса, то может оказаться, что некоторые кодовые слова имеют вес, отличный от веса других кодовых слов, принадлежащих тому же простому коду. Например,

$$A \cdot (13 \cdot 2^5 \bmod 105) = A \cdot 101 = (111101100011) = (10000\bar{1}0\bar{1}00\bar{1}0\bar{1})$$

и $W(A \cdot 101) = 5$. С другой стороны,

$$A \cdot (13 \cdot 2^{11} \bmod 105) = A \cdot 59 = (100011111101) = (10010000\bar{1}01)$$

и $W(A \cdot 59) = 4$. Однако, как видно из табл. 9.2, $A \cdot 101$ и $A \cdot 59$ получаются циклическим сдвигом $A \cdot 13$.

Причина этого «странного» явления, очевидно, состоит в том, что число $2^n - 1$ хотя и представляет 0, но имеет вес 2 (так как кодовые слова являются вычетами по модулю $2^n - 1$) [1]). Это явление возникает из-за того, что (00 ... 0) и (11 ... 1) используются для представления одного и того же кодового числа. Следовательно, если вернуться к приведенному выше примеру и из $A \cdot 101$ и $A \cdot 59$ вычесть $2^n - 1$, то полученные канонические представления будут иметь уже равный вес. Действительно, вычитая $2^n - 1 = (11 \ldots 1)$ из $A \cdot 101 = (111101100011)$, получаем кодовое слово $(0000\bar{1}00\bar{1}\bar{1}\bar{1}00)$ $= (0000\bar{1}0\bar{1}00100)$, которое является циклическим сдвигом $A \cdot 13 = (00100000\bar{1}0\bar{1})$. Другими словами, если циклические сдвиги канонического представления $A \cdot \alpha$ приводят к кодовому слову, являющемуся представлением отрицательного числа N, то, поскольку абсолютное значение N меньше, чем $2^n - 1$, сло-

[1]) Обратив на это внимание, Рао и Гарсиа [11] ввели понятия *модулярного веса* и *модулярного расстояния*. Они определили модулярный вес по модулю m целого числа I следующим образом:

$$W_m(I) = \min\{W(I), W(\ominus I)\},$$

где $W(\ominus I) = W(m - I) = W(I - m)$. При этом модулярное расстояние по модулю m между целыми числами I_1 и I_2 определяется равенством

$$D_m(I_1, I_2) = W_m(I_1 \ominus I_2),$$

где $I_1 \ominus I = (I_1 - I_2) \bmod m$. Для модулярного расстояния по модулю $m = 2^n - 1$ описанное выше «странное» явление не возникает. В то же время известно, что при любых m, равных $2^n - 1$, 2^n, $2^n + 1$, минимальное арифметическое расстояние AN-кодов, для которых $m = A \cdot B$, совпадает с минимальным модулярным расстоянием. Детально эти вопросы рассмотрены в работах [8] и [11]. (Эти вопросы подробно разбираются также в работе Г. А. Кабатянского [13]. — *Прим. ред.*)

жением N с $2^n - 1$ ($\equiv 0 \pmod{(2^n - 1)}$) можно получить положительное число $A \cdot (\alpha 2^j) \bmod B$. Вес этого числа можно оценить сверху следующим образом:

$$W(N + 2^n - 1) \leqslant W(N) + W(2^n - 1) = W(A\alpha) + 2.$$

В приведенном выше примере $W(A \cdot 13) = 3$ и $W(A \cdot 101) = 5 \leqslant W(A \cdot 13) + 2 = 5$, $W(A \cdot 59) = 4 < 5$.

Если выбрать α таким образом, что

$$\alpha < \alpha 2^j \bmod B,\ 0 < j \leqslant n - 1,$$

то, как следует из вышеизложенного, $W(A \cdot \alpha)$ дает минимальный вес кодовых слов простого кода, содержащего $A \cdot \alpha$. Таким образом, если определить вес такого кодового слова $A \cdot \alpha$, то можно оценить минимальное расстояние простого кода $C_\alpha(B)$.

Пусть задано двоичное представление $A \cdot \alpha$:

$$A \cdot \alpha = (a_{n-1} a_{n-2} \ \ldots \ a_0).$$

Тогда, так как $(2, B) = 1$, т. е. $B \equiv 1 \pmod 2$, и $-1 \equiv 1 \pmod 2$, то, заменяя b и e соответственно на a и n, формулу (9.9) можно переписать в виде

$$a_{n-i} \equiv \alpha 2^i \bmod B \equiv x_i \pmod 2, \tag{9.12}$$

где x_i — ненулевой вычет по модулю B.

Далее введем понятие центрального интервала C_B нечетного числа B, положив

$$C_B = \{x \bmod B \mid B \leqslant 3x < 2B\}. \tag{9.13}$$

Как показывает теорема 9.2, центральный интервал играет важную роль при оценке минимального расстояния AN-кодов.

Однако, прежде чем перейти к теореме 9.2, сформулируем в виде лемм два важных свойства центрального интервала [1]).

Лемма 9.1. *В C_B существует $((B + 1)/3)$ целых чисел, следующих подряд.*

Лемма 9.2. *Пусть $B \not\equiv 0 \pmod 3$. Тогда, если $x_i \in C_B$, то $(B - x_i) \in C_B$.* Другими словами, если $B < 3x_i < 2B$, то $B < 3(B - x_i) < 2B$.

Теорема 9.2. [2]). *Вес кодового слова $A \cdot \alpha = (a_{n-1}, a_{n-2}, \ldots, a_0)$ равен числу ненулевых, входящих в C_B вычетов x_i, таких, что*

$$a_{n-i} \equiv x_i = \alpha 2^i \bmod B \pmod 2,\ 1 \leqslant i \leqslant n. \tag{9.14}$$

---

[1]) Так как доказательства этих лемм являются простыми, то провести их читателям предлагается самостоятельно.

[2]) Эта теорема была независимо доказана Гото и Фукумура [4], а также Ченом, Хонгом и Препарата [5].

Доказательство. Рассмотрим двоичное представление числа $A \cdot 3\alpha$:

$$A \cdot 3\alpha = (b_{n-1}b_{n-2} \ldots b_0),$$

где

$$b_{n-i} \equiv 3\alpha \cdot 2^i \bmod B \equiv 3x_i \bmod B \pmod 2.$$

Используя алгоритм нахождения канонического представления, задаваемый соотношениями (8.50) и (8.51) (см. разд. 8.3), получаем

$$A \cdot 2\alpha = (d_{n-1}d_{n-2} \ldots d_0),$$

где $d_{n-i} = b_{n-i} - a_{n-i}$. Следовательно, вес $W(A \cdot 2\alpha)$ числа $A \cdot 2\alpha$ равен весу $W(A\alpha)$ числа $A\alpha$. Отсюда и из леммы 8.2 имеем

$$W(A \cdot 2\alpha) = W(A \cdot \alpha) = \sum_{i=1}^{n} |d_{n-i}|.$$

Докажем, что $|d_{n-i}| \neq 0$, а следовательно, и $a_{n-i} \neq b_{n-i}$ тогда и только тогда, когда $x_i = \alpha 2^i \bmod B$ является ненулевым вычетом, принадлежащим C_B. Отсюда непосредственно будет следовать справедливость теоремы.

Сначала докажем необходимость. Допустим, что $x_i \in C_B$. Так как по определению

$$B \leqslant 3x_i < 2B$$

и B является нечетным числом, то

$$b_{n-i} \equiv 3x_i \bmod B \pmod 2 \equiv 3x_i - B \pmod 2 \equiv$$
$$\equiv x_i \bmod 2 - 1 \pmod 2 \neq a_{n-i}.$$

Следовательно, $|d_{n-i}| \neq 0$.

Далее докажем достаточность. Предположим, что $x_i \notin C_B$. Тогда

а) если $3x_i < B$, то

$$b_{n-i} \equiv 3x_i \bmod B \pmod 2 \equiv$$
$$\equiv 3x_i \pmod 2 \equiv x_i \pmod 2 = a_{n-i};$$

б) если $2B \leqslant 3x_i < 3B$, то

$$b_{n-i} \equiv 3x_i - 2B \pmod 2 \equiv x_i \pmod 2 = a_{n-i}.$$

Таким образом, $d_{n-i} \neq 0$ тогда и только тогда, когда ненулевой вычет x_i принадлежит C_B и, следовательно, вес кодового слова $A\alpha$ равен числу таких вычетов $x_i = \alpha 2^i \bmod B$.

В заключение укажем необходимые и достаточные условия того, что код является *дополняющим*, а именно что для любого кодового слова $A\alpha = (a_{n-1}a_{n-2} \ldots a_0)$ при всех $j \leqslant n/2$ имеет место сравнение

$$a_{n-j} + a_{n-(j+n/2)} \equiv 1 \pmod 2. \qquad (9.15)$$

Выполнение сравнения (9.15) гарантирует, что оба кодовых слова $A\alpha$ и $-A\alpha$ принадлежат простому коду $C_\alpha(B)$. Это является следствием того, что кодовое слово $-A\alpha$ может быть получено из $A\alpha$ путем $n/2$ циклических сдвигов.

Если же код не является дополняющим, то для любого простого кода $C_\alpha(B)$ существует простой код $C_{-\alpha}(B)$. Эти два простых кода являются взаимно дополняющими в том смысле, что число, дополнительное к произвольному кодовому слову одного из этих кодов, является кодовым словом другого простого кода.

Лемма 9.3. *Двоичное представление* $A\alpha = (a_{n-1}a_{n-2} \ldots a_0)$ *числа* $A\alpha$ *обладает тем свойством, что* $a_{n-j} + a_{n-(j+n/2)} \equiv 1 \pmod 2$ *тогда и только тогда, когда* B *делит* $2^{n/2}+1$ (*где* $2^n - 1 = A \cdot B$).

Доказательство. Если $2^{n/2}+1 \equiv 0 \pmod B$, то

$$\begin{aligned} a_{n-j} + a_{n-(j+n/2)} &\equiv \alpha 2^j \bmod B + \alpha 2^{j+n/2} \bmod B \pmod 2 \equiv \\ &\equiv \alpha 2^j \bmod B + (-\alpha 2^j) \bmod B \pmod 2 \equiv \\ &\equiv \alpha 2^j \bmod B + (B - \alpha 2^j) \bmod B \pmod 2 \equiv \\ &\equiv 1 \pmod 2. \end{aligned}$$

Наоборот, если $a_{n-j} + a_{n-(j+n/2)} \equiv 1 \pmod 2$, $1 \leqslant j \leqslant n/2$, то при подходящем выборе целого числа k имеем $(2^{n/2}+1)A\alpha = k(2^n - 1)$. Так как $\alpha < B$, то B делит $2^{n/2}+1$.

9.2.2. Минимальное расстояние *AN*-кодов, удовлетворяющих специальным условиям

Минимальное расстояние произвольного циклического *AN*-кода можно оценить, используя результаты разд. 9.1 и 9.2.1. В этом разделе будет рассмотрено несколько случаев, когда для минимального расстояния кодов, удовлетворяющих определенным условиям, удается получить явные оценки.

Длина n, число кодовых слов B и порождающее число A всех рассматриваемых здесь кодов удовлетворяют условию $2^n - 1 = A \cdot B$, и, кроме того, $n = E(2, B)$ делит $\varphi(B)$. Все результаты, приведенные ниже в п. «а» — «ж», получаются непосредственно из лемм 9.1, 9.2 и теоремы 9.2.

а) $B = p$ (p — простое число, для которого число 2 является примитивным корнем).

Так как B — простое число, то в данном случае $n = \varphi(B) = B - 1$. Далее, поскольку 2 является примитивным корнем B, то существует $(B-1)$ ненулевых вычетов, порождающих мультипликативную группу $G(B)$, порождаемую числом 2. Поэтому, как уже указывалось в разд. 9.1.4, существует только одна под-

группа $S(B)$, соответствующая $G(B)$. Произвольное кодовое слово из $S(B)$ может быть получено путем циклических сдвигов $A \cdot 1 = (a_{n-1}a_{n-2}, \ldots, a_0)$ и имеет вес $[(B+1)/3]$.

Это можно показать следующим образом. Так как 2 является примитивным корнем B, то числа $x_i = 2^i \bmod B$ при изменении i от 1 до n «пробегают» все ненулевые вычеты. Следовательно, согласно лемме 9.1, в C_B входит $[(B+1)/3]$ вычетов. Отсюда и из теоремы 9.2 в свою очередь следует, что вес $A \cdot 1$ равен $((B+1)/3)$, а минимальное расстояние рассматриваемых AN-кодов равно

$$d_{\text{мин}} = \left[\frac{B+1}{3}\right]. \tag{9.16}$$

Далее, так как $2^n - 1 \equiv 0 \pmod B$ и $n = B - 1$ является четным числом, то $(2^{n/2} - 1)(2^{n/2} + 1) \equiv 0 \pmod B$. Если допустить, что $2^{n/2} - 1 \equiv 0 \pmod B$, то это противоречило бы тому, что $n = E(2, B)\,(= \varphi(B) = B - 1)$. Следовательно, $2^{n/2} + 1 \equiv 0 \pmod B$. Но тогда, согласно лемме 9.3, рассматриваемые коды являются самодополняющими.

Таблица 9.7

Циклический AN-код с $B = 13$

i	1	2	3	4	5	6	7	8	9	10	11	12
$2^i \bmod B$	2	4	8*	3	6*	12	11	9	5*	10	7*	1
$A \cdot 1$	0	0	0	1	0	0	1	1	1	0	1	1
$A \cdot 3$	0	0	1	1	1	0	1	1	0	0	0	1
$A \cdot 2$	0	0	1	0	1	0	0	0	$\bar{1}$	0	$\bar{1}$	0

В табл. 9.7 приведен код с $B = 13$ и $n = E(2, 13) = \varphi(13) = 12$. Так как $2^{12} - 1 = (3 \times 3 \times 5 \times 7) \times 13$, то $A = 315$.

Вес $A \cdot 2$, а следовательно, и вес $A \cdot 1$ можно определить по 3, 4 и 5-й строкам этой таблицы или по числу вычетов $x_i = 2^i \bmod B$, являющихся элементами $C_{13} = \{5, 6, 7, 8\}$ (в таблице эти вычеты отмечены знаком *).

б) $B = p$ — простое число, $n = \varphi(p)/2$ и n — нечетное число.

Так как в этом случае $2^{(p-1)/2} \equiv 1 \pmod p$, то 2 является квадратичным вычетом по модулю p. Следовательно, согласно формуле (8.24), $p = 8m \pm 1$, где m — целое число. Если $p = 8m - 1$, то $n = (p-1)/2$ — нечетное число. Таким образом, в данном случае $B = p$ является простым числом вида $8m - 1$. Если n — нечетное число, то, согласно лемме 9.3, код не является самодополняющим. Кроме того, заметим, что по условию $\varphi(p)/2 = n$. Следовательно, в $S(B)$ существуют два простых кода $C_1(B)$ и $C_{-1}(B)$. Однако, как следует из леммы 9.2, если

вычет $x_i = 2^i \bmod B$ принадлежит C_B, то $-x_i = -2^i \bmod B \in C_B$, а поэтому веса равномерно распределяются по $C_1(B)$ и $C_{-1}(B)$. Это означает, что минимальный вес равен $[(B+1)/3]/2$ и минимальное расстояние кода определяется равенством

$$d_{\text{мин}} = \frac{1}{2}\left[\frac{B+1}{3}\right]. \tag{9.17}$$

Аналогично определяется минимальное расстояние кодов, рассматриваемых ниже в п. «в»—«ж» [1]).

в) $B = p^a$ $(a > 1)$, p — простое число, 2 — примитивный корень p^a.

В этом случае

$$d_{\text{мин}} = \left\{\left[\frac{p^2+1}{3}\right] - \left[\frac{p+1}{3}\right]\right\} p^{a-2}. \tag{9.18}$$

В частности, если p имеет вид $3k+1$, то все подкоды имеют один и тот же минимальный вес, равный $1/3$ длины кода.

г) $B = p^a$ $(a > 1)$, p — простое число, 2 — квадратичный вычет по модулю p, $n = \varphi(p^a)/2$ — нечетное число.

$$d_{\text{мин}} = \frac{1}{2}\left\{\left[\frac{p^2+1}{3}\right] - \left[\frac{p-1}{3}\right]\right\} p^{a-2}. \tag{9.19}$$

д) $B = p_1 p_2$, p_1 и p_2 — простые числа, имеющие своим примитивным корнем число 2, B не делит $2^{n/2}+1$, $(p_1-1, p_2-1) = 2$.

$$d_{\text{мин}} = \frac{1}{2}\left\{\left[\frac{p_1p_2+1}{3}\right] - \left[\frac{p_1+1}{3}\right] - \left[\frac{p_2+1}{3}\right]\right\}. \tag{9.20}$$

е) $B = p_1 p_2$, p_1 и p_2 — простые числа, 2 — примитивный корень p_1 и квадратичный вычет по модулю p_2, B не делит $2^{n/2}+1$ и $(p_1-1, (p_2-1)/2) = 1$.

$$d_{\text{мин}} = \frac{1}{2}\left\{\left[\frac{p_1p_2+1}{3}\right] - \left[\frac{p_1+1}{3}\right] - \left[\frac{p_2+1}{3}\right]\right\}. \tag{9.21}$$

ж) $B = p_1^{\alpha_1} p_2^{\alpha_2}$ $(\alpha_1, \alpha_2 > 1)$, p_1 и p_2 — простые числа, 2 является примитивным корнем $p_1^{\alpha_1}$ и $p_2^{\alpha_1}$, $(p_1^{\alpha_1-1}(p_1-1), p_2^{\alpha_2-1}(p_2-1)) = 2$, B не делит $2^{n/2}+1$. Предположим, что $p_2 > p_1$.

1) Если $p_1 \equiv 1 \pmod 3$, $p_2 \equiv 2 \pmod 3$, то

$$d_{\text{мин}} = \frac{p_1^{\alpha_1-1}p_2^{\alpha_2-2}(p_1-1)}{2}\left\{\left[\frac{p_2^2+1}{3}\right] - \left[\frac{p_2+1}{3}\right]\right\}. \tag{9.22}$$

2) В остальных случаях

$$d_{\text{мин}} = \frac{p_2^{\alpha_2-1}p_1^{\alpha_1-2}(p_2-1)}{2}\left\{\left[\frac{p_1^2+1}{3}\right] - \left[\frac{p_1+1}{3}\right]\right\}. \tag{9.23}$$

[1]) Подробно эти вопросы рассматриваются в [2].

9.2.3. Минимальное расстояние циклических *AN*-кодов (*B* — простое число)

В разд. 9.2.2 были получены точные формулы для минимального расстояния кодов, удовлетворяющих одному из следующих двух условий. Первое условие состояло в том, что каждый из подкодов содержит лишь один простой код. Другое условие требует, чтобы длина кода была нечетной, а число простых кодов не превышало двух. При выполнении этих условий легко найти распределение весов, а следовательно, и минимальное расстояние кода.

В этом разделе рассматриваются оценки минимального расстояния циклических *AN*-кодов, параметр *B* которых является произвольным простым числом. Оценки минимального расстояния таких кодов представляют интерес не только сами по себе, но и используются для оценки минимального расстояния произвольных циклических кодов, для которых *B* является составным числом. Это связано с тем, что минимальное расстояние произвольного циклического *AN*-кода не может быть больше, чем минимальное расстояние какого-либо его подкода $S(p_i)$, параметр p_i которого является делителем *B*.

Общую формулу для минимального расстояния произвольного циклического *AN*-кода, число кодовых слов *B* которого является простым числом, явно записать не удается. Можно, однако, указать простой способ вычисления не только минимального расстояния, но и распределения весов.

Как уже неоднократно указывалось выше, длина *n* циклического *AN*-кода определяется из сравнения

$$2^n - 1 \equiv 0 \pmod{B}.$$

При этом $n = E(2, B)$ и *n* делит $\varphi(B) = B - 1$. Положим

$$\nu = \frac{B-1}{n}. \tag{9.24}$$

Из разд. 9.1.4 следует, что существует ν простых кодов. Все вычеты по модулю *B* образуют мультипликативную группу $G(B)$, которая может быть разложена на ν классов, каждому из которых соответствуют ν простых кодов $C_{a_i}(B)$, как указано в табл. 9.8. Вычеты α_i выбираются последовательно таким образом, чтобы α_i не принадлежал ни одному из i уже построенных классов.

Если множество $\{2^1, 2^2, \ldots, 2^{n-1}, 2^n \equiv 2^0\}$ рассматривать как подгруппу порядка *n* группы $G(B)$, то разложение, задаваемое табл. 9.8, будет не чем иным, как разложением $G(B)$ на смежные классы по подгруппе $\{2^1, 2^2, \ldots, 2^n\}$. Следовательно, если

Таблица 9.8

Последовательное разложение $G(B)$

$2^1 \ \dots \ 2^{n-1}$	$2^n \equiv 2^0 = 1$	$C_1(B)$
$\alpha_1 2^1 \ \dots \ \alpha_1 2^{n-1}$	α_1	$C_{\alpha_1}(B)$
. . .	.	.
. . .	.	.
. . .	.	.
$\alpha_{\nu-1} 2^1 \ \dots \ \alpha_{\nu-1} 2^{n-1}$	$\alpha_{\nu-1}$	$C_{\alpha_{\nu-1}}(B)$

g — примитивный корень B, то, выбирая элементы α_i следующим образом:

$$\begin{aligned} &\alpha_i = g^l \bmod B, \ 0 < i \leqslant \nu - 1, \\ &\alpha_0 = g^0 = 1, \end{aligned} \tag{9.25}$$

разложение, задаваемое табл. 9.8, можно построить регулярным способом, как показано в табл. 9.9. Разложение, полученное таким образом, и разложение в виде табл. 9.8, полученное последовательным выбором элементов α_i, могут отличаться порядком следования строк и порядком следования элементов в каждой строке, но тем не менее они определяют одно и то же разложение $G(B)$.

Таблица 9.9

Разложение $G(B)$ при выборе α_l с помощью формулы (9.25)

$1 = g^0$	$g^\nu \ \dots \ g^{\nu(n-1)}$	$C_1(B)$
g^1	$g^{\nu+1} \ \dots \ g^{\nu(n-1)+1}$	$C_{g^1 \bmod B}(B)$
. . .	. . .	.
. . .	. . .	.
. . .	. . .	.
$g^{\nu-1}$	$g^{2\nu-1} \ \dots \ g^{n\nu-1}$	$C_{g^{\nu-1} \bmod B}(B)$

Действительно, 2^k можно представить в виде $2^k \equiv g^j \pmod B$, где j — некоторое целое число, такое, что $j < B - 1$. Следовательно, так как $2^{kn} \equiv g^{jn} \equiv 1 \pmod B$, то $jn = l(B-1)$, где l — некоторое целое число. Отсюда и из формулы (9.24) получаем, что $j = l\nu$. Это означает, что 2^k имеет вид $g^{l\nu}$ и $l = 0$ только в том случае, когда $k = 0$. Покажем, что если $a, b < \nu$, то g^a и g^b не могут принадлежать одному и тому же смежному классу. Действительно, если бы эти элементы входили в один смежный класс, то, предполагая, что $b > a$, мы при-

шли бы к равенству $g^b = g^{a+lv}$ $(0 < l < v)$. Однако этого быть не может, так как $b > v$.

Таким образом, каждый простой код является одним из кодов $C_{a_i}(B)$, $a_i = g^i \bmod B$, и, согласно теореме 9.2, его минимальное расстояние равно числу вычетов $g^{i+jv} \bmod B$, $0 \leqslant j \leqslant n-1$, входящих в центральный интервал $C_B = \{x \bmod B \mid B \leqslant 3x < 2B\}$ числа B. Следовательно, определив минимальные расстояния всех простых кодов C_{a_i}, $i = 0, 1, \ldots, v-1$, и выбрав среди них минимальное, можно определить минимальное расстояние подкода $S(B)$, а следовательно, и исходного кода.

Пример 9.2. Определим минимальное расстояние циклического AN-кода с $B = 89$. Для этого кода $n = E(2, 89) = 11$, $v = (B-1)/n = 88/11 = 8$ и табл. 9.8 переходит в табл. 9.10.

Таблица 9.10

Вычисление минимального расстояния циклического AN-кода с $B = 89$ (таблица соответствует табл. 9.8)

i	1	2	3	4	5	6	7	8	9	10	11	Вес
$2^i \bmod 89$	2	4	8	16	32*	64	39*	78	67	45*	1	3
$3 \cdot 2^i \bmod 89$	6	12	24	48*	7	14	28	56*	23	46*	3	3
$5 \cdot 2^i \bmod 89$	10	20	40*	80	71	53*	17	34*	68	47*	5	4
$9 \cdot 2^i \bmod 89$	18	36*	72	55*	21	42*	84	79	69	49*	9	4
$11 \cdot 2^i \bmod 89$	22	44*	88	87	85	81	73	57*	25	50*	11	3
$13 \cdot 2^i \bmod 89$	26	52*	15	30*	60	31*	62	35*	70	51*	13	5
$19 \cdot 2^i \bmod 89$	38*	76	63	37*	74	59*	29	58*	27	54*	19	5
$33 \cdot 2^i \bmod 89$	66	43*	86	83	77	65	41*	82	75	61	33*	3

В этой таблице знаком * обозначены вычеты, входящие в множество $C_B = \{30, 31, 32, \ldots, 59\}$; число таких вычетов указывается в последнем столбце таблицы.

То же самое разложение можно получить с помощью табл. 9.9. В данном случае 6 является примитивным корнем $B = 89$ и табл. 9.9 переходит в табл. 9.11.

Табл. 9.10 и 9.11 дают одно и то же распределение весов: 3, 3, 4, 5, 3, 3, 4, 5. Следовательно, рассматриваемый код имеет минимальное расстояние $d_{\text{мин}} = 3$ и может исправлять любые ошибки веса 1.

В табл. 9.12 приведены значения минимального расстояния, вычисленные указанным выше образом для $B < 300$. Кроме минимального расстояния, в этой таблице приводятся также число простых кодов v, длина кода n, число исправляемых ошибок t и вычет $x_i = g^i \bmod B$, соответствующий простому коду $C_{x_i}(B)$, который дает минимальное расстояние.

Таблица 9.11

Определение минимального расстояния циклического *AN*-кода с $B = 89$ (таблица соответствует табл. 9.9)

j	Вычеты $6^j \bmod 89$											Вес
	0	1	2	3	4	5	6	7	8	9	10	
$6^{0+j\nu}$	1	8	64	67	2	16	39*	45*	4	32*	78	3
$6^{1+j\nu}$	6	48*	28	46*	12	7	56*	3	24	14	23	3
$6^{2+j\nu}$	36*	21	79	9	72	42*	69	18	55*	84	49*	4
$6^{3+j\nu}$	38*	37*	29	54*	76	74	58*	19	63	59*	27	5
$6^{4+j\nu}$	50*	44*	85	57*	11	88	81	25	22	87	73	3
$6^{5+j\nu}$	33*	86	65	75	66	83	41*	61	43*	77	82	3
$6^{6+j\nu}$	20	71	34*	5	40*	53*	68	10	80	17	47*	4
$6^{7+j\nu}$	31*	70	26	30*	62	51*	52*	60	35*	13	15	5

Таблица 9.12

Минимальное расстояние циклических *AN*-кодов, соответствующих простым B (< 300) и не относящимся к кодам, рассмотренным в разд. 9.2.2, а, б

B	ν	$d_{\text{мин}}$	x_i	t	n	B	ν	$d_{\text{мин}}$	x_i	t	n
17	2	2	1	0	8	157	3	14	5	6	52
31	6	1	1	0	5	193	2	28	1	13	96
41	2	6	1	2	20	223	6	10	1	4	37
43	3	4	1	1	14	229	3	22	1	10	76
73	8	2	1	0	9	233	8	8	9	3	29
89	8	3	1	1	11	241	10	4	1	1	24
97	2	14	1	6	48	251	5	14	1	6	50
109	3	10	6	4	36	257	16	2	1	0	16
113	4	6	1	2	28	277	3	28	1	13	92
127	18	1	1	0	7	281	4	20	9	9	70
137	2	20	1	9	68	283	3	28	9	13	94
151	10	4	1	1	15						

В заключение приведем верхнюю границу для минимального расстояния циклических AN-кодов, параметр B которых является простым числом. Вывод этой границы основан на том, что среднее арифметическое совокупности чисел является верхней границей для минимального числа в этой совокупности. Определим среднее значение минимального расстояния ν простых кодов. Так как общее число вычетов, входящих в C_B, равно $[(B+1)/3]$ и эти вычеты распределяются по ν простым кодам, то среднее

арифметическое минимального расстояния равно

$$\frac{1}{\nu}\left[\frac{B+1}{3}\right]=\frac{n}{B-1}\left[\frac{B+1}{3}\right]. \qquad (9.26)$$

Приближенно эта величина равна $n/3$. Эта граница справедлива не только для кодов, рассматривавшихся в этом разделе, но и является достаточно хорошей верхней границей минимального расстояния циклических AN-кодов, характеристики которых определяются подкодом $S(p)$ (p — простое число).

9.2.4. Минимальное расстояние циклических AN-кодов (B — составное число)

Минимальное расстояние кодов, для которых B является составным числом, определяется почти так же, как и в рассмотренном в разд. 9.2.3 случае простого B. Отличие состоит лишь в том, что если при простом B существует лишь один подкод $S(B)$, то при составном B необходимо рассматривать уже несколько мультипликативных групп $G(B/d)$, соответствующих подкодам $S(B)_d$, для различных делителей d числа B. Кроме того, если B не является простым числом, то $G(B)$ не имеет порождающего элемента, подобного элементу g, использованному в разд. 9.2.3.

Первое из этих отличий приводит лишь к увеличению числа подкодов, которые должны быть рассмотрены, и увеличению в связи с этим объема вычислений, но не создает каких-либо принципиальных трудностей.

Рассмотрим особенности определения минимального расстояния, связанные со вторым из указанных выше отличий. Так как B является нечетным числом, то множество степеней 2^k числа 2, взятых по модулю B, является подгруппой группы $G(B)$. Если разложить $G(B)$ по этой подгруппе на $\nu=\varphi(B)/n$ смежных классов, как было описано в разд. 8.1.8, то можно получить таблицу, аналогичную табл. 9.8 из предыдущего раздела, т. е.

Таблица 9.13

Классификация вычетов $G(B)$ (последовательное разложение)

$2^0\equiv 1$	2^1	2^2	$2^3 \dots 2^{n-1}$	$C_{\alpha_0}(B)$
α_1	$\alpha_1 2^1$	$\alpha_1 2^2$	$\alpha_1 2^3 \dots \alpha_1 2^{n-1}$	$C_{\alpha_1}(B)$
α_2	$\alpha_2 2^1$	$\alpha_2 2^2$	$\alpha_2 2^3 \dots \alpha_2 2^{n-1}$	$C_{\alpha_2}(B)$
.	.	.	. . .	.
.	.	.	. . .	.
.	.	.	. . .	.
$\alpha_{\nu-1}$	$\alpha_{\nu-1}2^1$	$\alpha_{\nu-1}2^2$	$\alpha_{\nu-1}2^3 \dots \alpha_{\nu-1}2^{n-1}$	$C_{\alpha_{\nu-1}}(B)$

расположить вычеты входящие в $G(B)$, так, как показано в табл. 9.13. Лидер α_i $(0 \leqslant i \leqslant v-1)$ каждого смежного класса этой таблицы выбирается из элементов, не вошедших ни в один из смежных классов, порождаемых предшествующими лидерами α_j, $j < i$ $(\alpha_0 = 1)$. Согласно теореме 9.2, минимальное расстояние простого кода $C_{\alpha_i}(B)$ определяется числом вычетов $\alpha_i \cdot 2^k \bmod B$, $0 \leqslant k \leqslant n-1$, лежащих в центральном интервале $C_B = \{x \bmod B \mid B \leqslant 3x < 2B\}$ числа B. Следовательно, вычислив минимальные расстояния всех простых кодов $C_{\alpha_i}(B)$ и выбрав из них минимальное, можно определить минимальное расстояние $S(B)$.

Вычисленные таким образом значения минимального расстояния некоторых циклических AN-кодов, параметр B которых является составным числом, приведены в табл. 9.14.

Таблица 9.14

Минимальное расстояние $d_{\text{мин}}$ и длина $n = E(2, B)$ циклических AN-кодов с составным B (< 150)

B	n	$d_{\text{мин}}$	B	n	$d_{\text{мин}}$	B	n	$d_{\text{мин}}$
15	4	1	77	30	10	117	12	3
21	6	2	85	8	2	119	24	6
33	10	2	87	28	9	123	20	6
35	12	4	91	12	3	129	14	2
39	12	4	93	10	2	133	18	5
45	12	3	95	36	12	135	36	9
51	8	2	99	30	6	141	46	15
57	18	4	105	12	3	143	60	20
69	22	7	111	36	12	145	28	6
75	20	5	115	44	14	147	42	14

В заключение этого раздела остановимся на связи между разложением чисел вида $2^n - 1$ на множители и построением различных циклических AN-кодов, имеющих одну и ту же длину n. Обычно в разложение числа

$$2^n - 1 = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_k^{\alpha_k} = A \cdot B \tag{9.27}$$

входит несколько простых делителей. Распределяя эти делители между A и B так, чтобы выполнялось условие $n = E(2, B)$, всегда будем получать коды одной и той же длины n. Как было указано в разд. 8.1.11, число $2^n - 1$ может быть представлено в виде произведения круговых чисел, а именно

$$2^n - 1 = \prod_{d \mid n} \Phi_d(2), \tag{9.28}$$

Если среди делителей B существует простое число, большее n, которое делит $\Phi_n(2)$, то $n = E(2, B)$. Если B имеет такой сомножитель, то независимо от того, каковы другие сомножители B, получающийся код будет иметь длину n. Следовательно, выбирая B по возможности большим, но так, чтобы выполнялось указанное выше условие, можно максимизировать число кодовых слов в коде, не увеличивая длину последнего. Однако вполне естественно, что при этом может уменьшиться минимальное расстояние. Как можно видеть из формулы (9.28), число различных делителей $2^n - 1$ не меньше числа делителей у n. Поэтому всегда можно построить циклические AN-коды одной и той же длины с различным числом кодовых слов и различным минимальным расстоянием.

Например, рассмотрим разложение $2^{28} - 1$:

$$2^{28} - 1 = \Phi_{28}(2)\,\Phi_{14}(2)\,\Phi_7(2)\,\Phi_4(2)\,\Phi_2(2)\,\Phi_1(2) =$$
$$= (29 \times 113) \times 43 \times (127) \times (5) \times (3) \times (1).$$

Некоторые возможные способы распределения простых делителей между A и B приведены в табл. 9.15.

Таблица 9.15

Распределение делителей $2^{28} - 1$ и минимальное расстояние получающихся кодов

A	B	$d_{\text{мин}}$
1	$29 \times 3 \times 5 \times 43 \times 113 \times 127$	1
127	$29 \times 3 \times 5 \times 43 \times 113$	2
127×113	$29 \times 3 \times 5 \times 43$	4
$127 \times 113 \times 43$	$29 \times 3 \times 5$	6
$127 \times 113 \times 43 \times 5$	29×3	9
$127 \times 113 \times 43 \times 5 \times 3$	29	10

Все задаваемые табл. 9.15 коды содержат в качестве подкода S (29). Так как 2 является примитивным корнем 29, то минимальное расстояние подкода $S(29)$ равно $[(29 + 1)/3] = 10$. Как видно из табл. 9.15, по мере введения в число сомножителей B других делителей $2^{28} - 1$, отличных от 29, число кодовых слов возрастает, а минимальное расстояние $d_{\text{мин}}$ уменьшается. В этой таблице в качестве сомножителя B, который делит $\Phi_{28}(2)$, используется число 29. Поскольку $E(2, 113) = E(2, 29) = 28$, то вместо 29 можно использовать также и 113. Однако минимальное расстояние $S(113)$ равно 6, а поэтому минимальное расстояние кодов, параметр B которых имеет в качестве сомножителя 113, не может превышать 6.

Наконец, заметим, что если отказаться от того, что B должно содержать в качестве одного из своих сомножителей простое число, которое делит $\Phi_n(2)$, то, определив все делители B числа $2^n - 1$, удовлетворяющие условию $n = E(2, B)$, можно найти максимально возможное значение B, порождающее циклический AN-код заданной длины n с заданным минимальным расстоянием d. Этот код имеет наибольшее число кодовых слов из всех кодов заданной длины с заданным минимальным расстоянием и в этом смысле является оптимальным.

9.3. Декодирование циклических AN-кодов

Так как при декодировании арифметических кодов все вычисления выполняются в поле действительных чисел, то обнаружение и исправление ошибок легко можно реализовать с помощью универсальной вычислительной машины, разработав для нее специальную программу. Это является одним из достоинств арифметических кодов. Арифметические коды могут исправлять ошибки, возникающие в арифметических устройствах, при передаче данных между вычислительными машинами, во входных и выходных устройствах, и другие ошибки, результатом которых является арифметическая ошибка. Декодер арифметического кода рассматривает все ошибки независимо от причин, породивших их, как арифметические, и если эти ошибки лежат в области корректирующей способности используемого кода, то он правильно восстанавливает исходное кодовое слово. В разд. 8.7 был описан метод декодирования произвольных арифметических кодов по синдрому. В этом разделе будут описаны два эффективных алгоритма декодирования циклических AN-кодов, использующие специальные свойства последних.

9.3.1. Вычеты по модулю A и конфигурации ошибок

Тогда как при построении и вычислении минимального расстояния циклических AN-кодов важную роль выполняют вычеты по модулю B, при исправлении ошибок более важными оказываются вычеты по модулю A. Установим сначала связь между ошибками и вычетами по модулю A, а также укажем некоторые свойства ошибок.

Ошибка всегда может быть задана с помощью двоичного представления целого числа E:

$$E = \sum_{i=0}^{n-1} c_i 2^i, \quad c_i \in \{0, 1\} \tag{9.29}$$

(здесь n — длина кода). Ошибка $2^j E$ получается сдвигом двоичного представления ошибки E на j разрядов влево и имеет ту

же конфигурацию, что и E. Здесь и ниже конфигурациями ошибок называются двоичные представления нечетных чисел E, меньших A [1]). Зададим конфигурацию $E_0 \neq 0$ и будем рассматривать только такие ошибки, которые представляются в виде $2^j E_0$ $(0 \leqslant j \leqslant n-1)$. При рассмотрении AN-кодов с минимальным расстоянием d ошибки E веса $W(E) \leqslant (d-1)/2$ будем называть исправляемыми ошибками. Иначе говоря, исправляемыми ошибками называются ошибки E, вес $W(E)$ которых не превышает корректирующей способности кода. Далее, если конфигурация ошибок E_0 удовлетворяет условию $W(E_0) \leqslant (d-1)/2$, то она будет называться исправляемой конфигурацией ошибок.

Если в процессе передачи в арифметическом устройстве возникает ошибка, в результате которой вместо кодового слова αA $(\alpha < B)$ принимается другое слово, то последнее всегда можно представить в виде

$$\alpha A + 2^j E_0. \tag{9.30}$$

Взяв остаток от деления числа, соответствующего этому слову, на A, получим

$$r_A(\alpha A + 2^j E_0) = (\alpha A + 2^j E_0) \bmod A = 2^j E_0 \bmod A. \tag{9.31}$$

Этот остаток зависит только от ошибки $2^j E_0$ и, как указывалось в разд. 8.7, часто называется синдромом. Если синдром равен нулю, то ошибка не обнаруживается (это имеет место, если ошибки не возникают или если возникают необнаруживаемые ошибки). Если синдром отличен от нуля, то это означает, что произошла ошибка.

Чтобы исправить эту ошибку, необходимо определить конфигурацию ошибки E_0 и число j, указывающее место, в котором она возникла.

Следующая лемма устанавливает взаимно однозначное соответствие между ошибками $2^j E_0$ и вычетами по модулю A.

Лемма 9.4. *Если E_0 является исправляемой конфигурацией ошибок, т. е. если*

$$W(E_0) \leqslant (d-1)/2,$$

то ни при каком j, $0 < j < n$, вычет

$$R = 2^j E_0 \bmod A$$

не совпадает с E_0. Кроме того, если $i \neq j$, $0 < i, j < n$, то

$$2^i E_0 \bmod A \neq 2^j E_0 \bmod A.$$

---

[1]) Особой конфигурацией ошибок является конфигурация $E_0 = 0$, соответствующая отсутствию каких-либо ошибок.

Доказательство. Поскольку $R = 2^j E_0 \bmod A$, то

$$R = 2^j E_0 - \alpha A.$$

Если $R = E_0$, то

$$2^j E_0 - E_0 = \alpha A.$$

Так как минимальное расстояние кода равно d, то $W(\alpha A) \geqslant d$ для всех $0 < \alpha < B$. Следовательно,

$$W(2^j E_0 - E_0) \geqslant d.$$

Однако $W(2^j E_0 - E_0) \leqslant W(2^j E_0) + W(E_0) \leqslant d - 1$, и мы получили противоречие. Таким образом, равенство $R = E_0$ выполняться не может. Если $\alpha = 0$, то

$$2^j E_0 - E_0 \equiv 0 \pmod{A}.$$

Так как $j < n$, то E_0 и A не являются взаимно простыми. Пусть $(E_0, A) = g$. Тогда

$$2^j \equiv 1 \pmod{A/g}.$$

Отсюда можно найти минимальное значение j, удовлетворяющее приведенному выше сравнению. Так как в этом случае ошибка $2^j E_0$ совпадает с конфигурацией ошибки E_0, то вычет $R = 2^j E_0 \bmod A$ может совпадать с E_0. Вторая часть леммы доказывается аналогично.

Пример 9.3. Циклический AN-код длины $n = 18$, содержащий $B = 19$ кодовых слов ($A = 13797$), является кодом, исправляющим ошибки веса 2. Рассмотрим следующую исправляемую конфигурацию ошибок:

000000001000000001.

Эта конфигурация является двоичным представлением числа 513. Легко проверить, что $(513, A) = 27$. Ошибка $2^9 \cdot 513$ совпадает с приведенной выше конфигурацией и $2^9 \cdot 513 \equiv 513 \pmod{A}$.

Лемма 9.5. *Если исправляемая конфигурация ошибок* E_0 $(W(E_0) \leqslant (d-1)/2)$ *удовлетворяет условию* $2^j E_0 > A$, *то вычет* $2^j E_0 \bmod A$ *исправляемой ошибкой не является.*

Доказательство. Если предположить, что $R = 2^j E_0 \bmod A$, то при подходящем выборе $\alpha > 0$ будет иметь место равенство

$$2^j E_0 - R = \alpha A.$$

Следовательно,

$$W(2^j E_0 - R) = W(\alpha A) \geqslant d.$$

Однако так как $W(2^j E_0) + W(R) \geqslant W(2^j E_0 - R)$, то $W(2^j E_0) + W(R) \geqslant d$

и

$$W(R) \geqslant d - W(2^j E_0).$$

Таким образом, для веса R мы получили следующую верхнюю границу:

$$W(R) \geqslant d - \frac{d-1}{2} = \frac{d+1}{2}.$$

Эта граница показывает, что $R = 2^j E_0 \bmod A$ не является исправляемой ошибкой.

Пример 9.4. В табл. 9.16 приведены результаты вычисления вычетов $2^j E_0 \bmod A$ для кода из примера 9.3 и конфигурации ошибок $E_0 = 3$, $W(E_0) = 2$. Эта таблица служит хорошим примером к лемме 9.5.

Таблица 9.16

Циклические сдвиги конфигурации ошибок $E_0 = 0000000000000011$ и вес соответствующих вычетов

$R = 2^j E_0 \bmod A$

j	$R = 2^j E_0 \bmod A$	$W(R)$		j	$R = 2^j E_0 \bmod A$	$W(R)$	
0	3	2		9	1 536	2	
1	6	2		10	3 072	2	
2	12	2		11	6 144	2	$2^j E_0 < A$
3	24	2		12	12 288	2	
4	48	2	$2^j E_0 < A$				
5	96	2		13	10 779	7	
6	192	2		14	7 761	5	
7	384	2		15	1 725	5	$2^j E_0 > A$
8	768	2		16	3 450	5	
				17	6 900	5	

9.3.2. Метод декодирования, основанный на циклических сдвигах вычетов

В этом разделе описывается алгоритм декодирования, использующий циклический сдвиг вычетов. Сначала заметим, что если $R \equiv E2^j \pmod A$, то

$$2^{n-j} R \equiv E \pmod A,$$

так как $2^n - 1 \equiv 0 \pmod A$. Отсюда и из леммы 9.5 получаем следующий алгоритм декодирования. Сначала вычисляются вычет $R_0 = 2^j E_0 \bmod A$ по модулю A принятого кодового слова $\alpha A + 2^j E_0$ и его вес $W(R_0)$. Если $W(R_0)$ не превышает $(d-1)/2$, то R_0 является ошибкой. Если же $W(R_0) > (d-1)/2$, то R_0 сдвигается влево на один разряд, находится вычет $R_1 = 2R_0 \bmod A$ полученного в результате сдвига числа $2^1 R_0$ и его вес $W(R_1)$ сравнивается с $(d-1)/2$. Если $W(R_1) \leqslant (d-1)/2$,

то R_1 является конфигурацией ошибок. Тогда, осуществляя сдвиг на один разряд вправо, можно определить положение ошибки. Если $W(R_1) > (d-1)/2$, то осуществляется новый сдвиг, вычисляется вес полученного вычета по модулю A, принимается решение о том, принадлежит или нет этот вес области корректирующей способности, и т. д. Если после некоторого сдвига вычисленный вес оказывается лежащим в области корректирующей способности, то это означает, что конфигурация ошибок найдена и дальнейшие сдвиги влево не производятся. При этом определяется число сдвигов, потребовавшихся для нахождения конфигурации ошибок, и путем осуществления такого же числа сдвигов в обратном направлении устанавливается положение ошибки. Если ни один из сдвигов не приводит к вычету, вес которого лежит в области корректирующей способности кода, то это означает, что произошла обнаруживаемая, но не исправляемая данным кодом ошибка. Заметим, что некоторые неисправляемые ошибки могут декодером не обнаруживаться и декодироваться ошибочно.

Пример 9.5. Вновь рассмотрим циклический AN-код с параметрами $n = 18$ и $B = 19$, использованный ранее в примерах 9.3 и 9.4. Этот код исправляет ошибки веса 2. Предположим, что вычет по модулю A ($= 13\,797$) принятого кодового слова равен 7761. Его вес равен 5 и превышает 2. Поэтому, согласно описанному алгоритму декодирования, будем сдвигать этот вычет влево, вычисляя после каждого сдвига вычет по модулю A и его вес. Результаты этих вычислений приведены в табл. 9.17,

Таблица 9.17

R_k и $W(R_k)$ при $R_0 = 7761$

k	R_k	$W(R_k)$
0	7761	5
1	1725	5
2	3450	5
3	6900	5
4	3	2

где R_0 — вычет по модулю A принятого кодового слова, а R_k — вычет по модулю A после k сдвигов. Так как $W(R_4) = 2$ лежит в области корректирующей способности кода, то 5-й и последующие сдвиги проводить не нужно. Конфигурация ошибок в данном случае имеет вид

000000000000000011.

Осуществляя четыре сдвига в обратном направлении, находим ошибку

$$0011000000000000000.$$

Вычитая эту ошибку из принятого кодового слова, получаем исходное кодовое слово.

Если бы в действительности возникла ошибка 7761, имеющая двоичное представление

$$E = 000001111001010001$$

и вес которой превышает корректирующую способность кода, то при декодировании принятого кодового слова с помощью описанного выше алгоритма декодирования было бы получено то же кодовое слово, что и ранее, т. е. произошла бы ошибка декодирования.

Однако, как уже указывалось в разд. 8.3, арифметические ошибки бывают двух типов: ошибки сложения и ошибки вычитания. Ошибки сложения могут быть исправлены с помощью описанного выше алгоритма, но в действительности существуют еще ошибки вычитания. Хотя ошибка вычитания E является отрицательным числом, вычет по модулю A принятого кодового слова по определению всегда положителен. Действительно, вычет определяется числом $A - E$ или полученным в результате его циклических сдвигов числом $2^j(A - E) \bmod A = A - 2^jE \bmod A$. Поэтому для каждого положительного вычета $2^jE \bmod A$, соответствующего ошибке 2^jE, существует вычет $A - 2^jE \bmod A$, соответствующий ошибке $2^j(-E)$. Первый вычет будем обозначать через R^+, а второй — через R^-. Очевидно, что R^+ и R^- связаны равенством $R^- = A - R^+$.

Например, в табл. 9.18 приведены вычеты R^+ и R^- для кода из примера 9.5 и ошибок $E = \pm 3$.

Таблица 9.18

Вычеты $R^+ = 2^jE \bmod A$ и $R^- = 2^j(-E) \bmod A = A - R^+$

j	$R^+ = 2^jE \bmod A$	$R^- = A - R^+$	j	$R^+ = 2^jE \bmod A$	$R^- = A - R^+$
0	3	13 794	9	1 536	12 261
1	6	13 791	10	3 072	10 725
2	12	13 785	11	6 144	7 653
3	24	13 773	12	12 288	1 509
4	48	13 749	13	10 779	3 018
5	96	13 701	14	7 761	6 036
6	192	13 605	15	1 725	12 072
7	384	13 413	16	3 450	10 347
8	768	13 029	17	6 900	6 897

Таким образом, если задан только вычет R_0, то сначала неизвестно, соответствует ли он ошибке сложения или ошибке вычитания. Поэтому алгоритм декодирования должен включать вычисление $\bar{R}_0 = A - R_0$ и определение весов вычетов в процессе проведения циклических сдвигов как для R_0, так и для $\bar{R}_0$. Если вес оказывается в области корректирующей способности кода, т. е. если

$$W(R_k) \leqslant (d-1)/2,$$

то ошибка является ошибкой сложения. Если проведение циклических сдвигов заканчивается, когда в области корректирующей способности кода оказывается $W(\bar{R}_k)$, то ошибка является ошибкой вычитания. В последнем случае для восстановления исходного кодового слова найденная ошибка должна быть сложена с принятым кодовым словом.

Таким образом, при использовании описанного выше алгоритма декодирования необходимы веса как вычета R_k, так и дополнительного вычета $\bar{R}_k = A - R_k$. Для вычисления весов можно использовать метод, описанный в разд. 8.3. При вычислении весов полезно знать следующее. Так как вес вычета 2^jR_0 совпадает с весом вычета R_0, если только $2^jR_0 < A$, необязательно вычислять $W(2^jR_0)$. Однако, как легко видеть, если $2R_0 < A$, то $2\bar{R}_0 > A$. Верно и обратное. Следовательно, после каждого циклического сдвига достаточно вычислить вес лишь одного из вычетов R_k или $\bar{R}_k$ в зависимости от того, какое из неравенств $2R_{k-1} > A$ или $2R_{k-1} < A$ имеет место. Само собой разумеется, что если $W(R_0)$ или $W(\bar{R}_0)$ оказывается в области корректирующей способности кода, то необходимости в проведении дальнейших циклических сдвигов нет. При этом искомой конфигурацией ошибки является двоичное представление R_0 или $\bar{R}_0$ в зависимости от того, какое из неравенств $W(R_0) \leqslant (d-1)/2$ или $W(\bar{R}_0) \leqslant (d-1)/2$ имеет место. Если $W(R_0) \leqslant (d-1)/2$, то возникшая ошибка является ошибкой сложения, в противном случае — ошибкой вычитания.

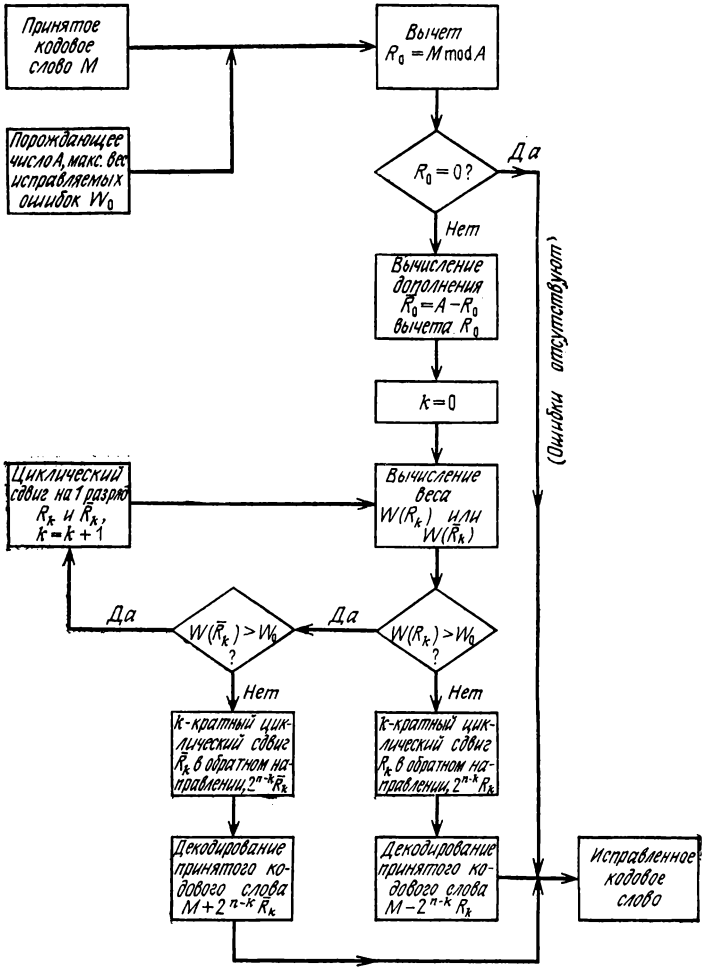
На фиг. 9.3 приведена блок-схема алгоритма декодирования, детально описанного выше.

9.3.3. Метод декодирования, основанный на переборе

Более простым методом декодирования по сравнению с тем, что был описан в предыдущем разделе, является метод декодирования, описанный ниже. Этот метод предусматривает вычисление расстояний между принятым словом и кодовыми словами и основан на том, что если расстояние между принятым словом и некоторым кодовым словом оказывается меньше или равным $(d-1)/2$, где d — минимальное расстояние используемого

кода, то это кодовое слово является искомым кодовым словом.

Для реализации этого метода необходимо иметь таблицу, в которой хранились бы все кодовые слова, или порождать кодо-



Фиг. 9.3. Алгоритм декодирования, основанный на циклических сдвигах вычетов.

вые слова некоторым способом. Этот метод применим не только для циклических AN-кодов, но и для произвольных AN-кодов и позволяет исправлять все ошибки E, вес которых не превосходит $(d-1)/2$. Однако в случае циклических AN-кодов последова-

тельное построение кодовых слов может быть реализовано сравнительно просто.

Для того чтобы декодировать этим методом одно принятое слово, необходимо провести сравнение этого слова приблизительно с половиной всех возможных кодовых слов. Более того, все эти сравнения необходимо проводить даже в том случае, если ошибок не было. Чтобы избежать этого, можно сначала вычислить вычет принятого кодового слова по модулю A, провести таким образом обнаружение ошибок и лишь после этого приступать к их исправлению. Приводимая ниже лемма гарантирует, что при использовании указанного алгоритма исправляются все указанные выше ошибки. Эта лемма утверждает, что если расстояние между принятым словом и некоторым кодовым словом не превышает $(d-1)/2$, то это кодовое слово является искомым кодовым словом и никакое другое кодовое слово не лежит ближе к принятому слову.

Лемма 9.6. *Пусть* $\alpha_1 A$ *и* $\alpha_2 A$ *— два различных кодовых слова* $(\alpha_1 \neq \alpha_2)$. *Если принято слово* $\alpha A + E$,

$$W(E) \leqslant \frac{d-1}{2}$$

и d *— минимальное расстояние кода, то неравенства*

$$d(\alpha A + E, \alpha_1 A) \leqslant (d-1)/2,$$

$$d(\alpha A + E, \alpha_2 A) \leqslant (d-1)/2$$

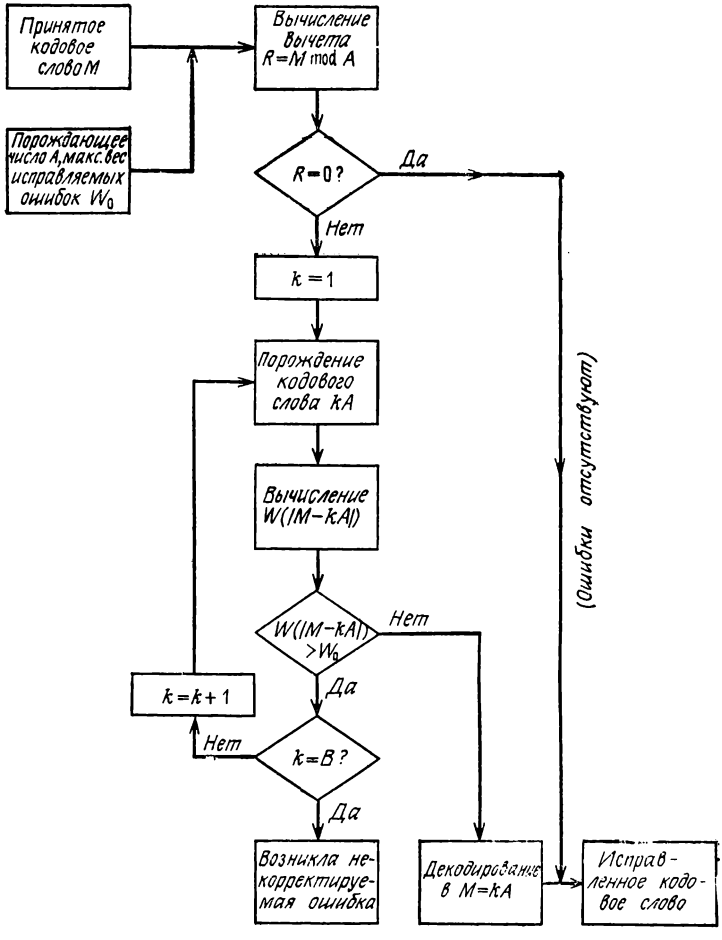
одновременно выполняться не могут. Кроме того, если выполняется первое из этих неравенств, то $\alpha A = \alpha_1 A$.

Доказательство. Если оба приведенных выше неравенства выполняются одновременно, то

$$\begin{aligned} d(\alpha_1 A, \alpha_2 A) &= W(|\alpha_1 A - \alpha_2 A|) = \\ &= W(|\alpha_1 A - \alpha A - E - \alpha_2 A + \alpha A + E|) = \\ &= W(|\alpha_1 A - \alpha A - E) - (\alpha_2 A - \alpha A - E)|) \leqslant \\ &\leqslant W(|\alpha_1 A - \alpha A - E|) + W(|\alpha_2 A - \alpha A - E|) = \\ &= d(\alpha A + E, \alpha_1 A) + d(\alpha A + E, \alpha_2 A) \leqslant d - 1. \end{aligned}$$

Однако так как код имеет минимальное расстояние d и $\alpha_1 \neq \alpha_2$, то $d(\alpha_1 A, \alpha_2 A) \geqslant d$, и мы получили противоречие. Следовательно, если $\alpha_1 \neq \alpha_2$, то приведенные в формулировке леммы два неравенства одновременно выполняться не могут. Справедливость равенства $\alpha A = \alpha_1 A$ при условии, что $d(\alpha A + E, \alpha_1 A) \leqslant (d-1)/2$ и $W(E) \leqslant (d-1)/2$, доказывается аналогично.

На фиг. 9.4 приведена блок-схема описанного выше алгоритма декодирования, в котором, однако, предусмотрено обнаружение ошибок путем вычисления вычета по модулю A для принятого слова.



Фиг. 9.4. Алгоритм декодирования, основанный на переборе.

9.4. Дополнение

В этой и предыдущей главах были описаны AN-коды, которые могут исправлять ошибки, возникающие в арифметических устройствах. Завершением этих глав является данное дополнение.

До сих пор рассматривались AN-коды, предназначенные для исправления независимых ошибок. Эти коды могут использоваться также и для исправления пачек ошибок, но их эффективность в этом случае будет невысокой. Действительно, так как A не делит никакой сдвиг 2^jE, где E — произвольная пачка ошибок длины $b = [\log_2 A]$ или менее, то этим кодом всегда могут быть обнаружены произвольные конфигурации пачки ошибок длины $b = [\log_2 A]$ и менее.

Далее, так как число ненулевых членов канонического представления, описанного в разд. 8.3, определяет минимальный вес, для максимального веса пачки ошибок E длины b имеет место следующая верхняя граница:

$$W(E) \leqslant \left[\frac{b}{2}\right] + 1. \tag{9.32}$$

Следовательно, произвольный циклический код, исправляющий $[b/2] + 1$ независимых ошибок, может исправлять пачки ошибок длины b. Другими словами, циклический AN-код с минимальным расстоянием d может исправлять любые пачки ошибок длины b и менее, где b определяется равенством

$$b = \begin{cases} d-2, & d \text{ — нечетное,} \\ d-3, & d \text{ — четное.} \end{cases} \tag{9.33}$$

AN-коды, обнаруживающие пачки ошибок и подобные кодам Файра, были предложены Мандельбаумом [6]. Эти коды имеют порождающие числа A вида

$$A = (2^c - 1)p. \tag{9.34}$$

При выполнении указанных ниже трех условий всегда могут быть обнаружены произвольные пачки ошибок $E = 2^iE_1 + 2^jE_2$ веса 2.

1) Длины b_1 и b_2 конфигураций пачек ошибок E_1 и E_2 удовлетворяют неравенству

$$c - 1 \geqslant b_1 + b_2. \tag{9.35}$$

2) Число p является простым и

$$[\log_2 p] \geqslant \min(b_1, b_2). \tag{9.36}$$

3) Длина кода $n \leqslant \text{НОК } (c, e)$, где $e = E(2, p)$.

Если c — нечетное число, $c \geqslant 2b$, $[\log_2 p] = b$, $n \leqslant$ НОК (c, e), то коды с такими параметрами могут обнаруживать произвольные пачки ошибок веса 2, длины b и менее. Кроме того, эти коды могут исправлять произвольные пачки ошибок веса 1, длины b и менее.

Гото и Фукумура [7] установили необходимые и достаточные условия того, что код исправляет пачки ошибок веса 1 или обнаруживает пачки ошибок веса 2. Они показали, что этим условиям удовлетворяют AN-коды, исправляющие ошибки веса 1 с порождающими числами вида $p(2^c \pm 1)$, $p(2^{2km} - 1)/(2^k + 1)$, $p(2^{k(2m-1)} + 1)/(2^k + 1)$, где p — простое число.

К числу кодов, используемых в арифметических устройствах, относятся также как называемые *многоостаточные коды*. Многоостаточный код определяется k целыми положительными числами $m_1, m_2, \ldots, m_k$ и сопоставляет каждому числу $N (0 \leqslant N \leqslant m - 1, m = 2^n - 1)$ совокупность $(k + 1)$ целых чисел $(N, C_1(N), C_2(N), \ldots, C_k(N))$, где $C_i(N)$ — вычет N по модулю m_i и m_i — положительный делитель m. Кодовое слово такого кода представляет собой последовательность двоичных представлений числа N и вычетов $C_1(N), C_2(N), \ldots, C_k(N)$. Так как

$$C_i(N_1 \pm N_2) = (N_1 \pm N_2) \bmod m_i = (C_i(N_1) \pm C_i(N_2)) \bmod m_i \quad (9.37)$$

и

$$C_i(N_1 \cdot N_2) = (N_1 \cdot N_2) \bmod m_i = (C_i(N_1) \cdot C_i(N_2)) \bmod m_i, \quad (9.38)$$

то сумма, разность и произведение кодовых слов также являются кодовыми словами. Отметим также, что в кодовых словах многоостаточных кодов можно выделить информационные символы, представляющие число N, и проверочные символы, представляющие k остатков $C_i(N)$, что является одним из достоинств этих кодов.

Между многоостаточными кодами и AN-кодами существует тесная связь. Многоостаточному коду, порождаемому целыми числами $m_1, m_2, \ldots, m_k$, можно поставить в соответствие AN-код, взяв в качестве порождающего числа последнего НОК $(m_1, m_2, \ldots, m_k)$. Число кодовых слов B этого AN-кода определяется из равенства $AB = m = 2^n - 1$. При этом по k вычетам $C_1(N), \ldots, C_k(N)$ произвольного числа $N (0 \leqslant N \leqslant m - 1)$ однозначно может быть определен вычет $N \bmod A$ по модулю A и, наоборот, по вычету $N \bmod A$ могут быть определены однозначно вычеты $C_1(N), \ldots, C_k(N)$ [1]. В силу этого многоостаточные коды и соответствующие им AN-коды имеют одинаковую корректирующую способность.

---

[1]) Это является следствием теоремы теории целых чисел, известной под названием китайской теоремы об остатках.

Опубликовано несколько работ, посвященных многоостаточным кодам [8, 11], в которых затронутые выше вопросы рассмотрены более детально.

На большое сходство AN-кодов и циклических алгебраических кодов уже указывалось в конце разд. 8.5. Однако если рассматривать циклические AN-коды, то это сходство оказывается еще более глубоким. Этот вопрос подробно рассматривается в работе Месси и Гарсиа [8]. В последние годы было проведено несколько исследований, посвященных поиску циклических AN-кодов, соответствующих некоторым хорошим циклическим алгебраическим кодам и исследованию их свойств [1]).

В заключение кратко остановимся на применении рассмотренных кодов. В настоящее время для повышения надежности вычислительных устройств применяются следующие методы: 1) использование трех идентичных устройств (тройное дублирование) для выполнения одной и той же операции и мажоритарное декодирование их выходов; 2) использовние двух идентичных устройств (двойное дублирование) с целью обнаружения неисправностей и 3) использование арифметических устройств, предусматривающее кодирование подлежащей обработке информации и контроль последней после обработки. Последний из этих методов не гарантирует исправления ошибок при всех возможных арифметических операциях, но стоимость аппаратуры, необходимой для его реализации, меньше, чем стоимость дополнительных устройств, необходимых для реализации первого и второго методов. Кроме того, использование арифметических кодов позволяет проводить одними и теми же методами исправление ошибок, возникающих как при передаче данных, так и при их обработке в арифметических устройствах. Благодаря этому арифметические коды эффективно могут использоваться в современных системах связи и других устройствах.

Наиболее глубоким исследованием, посвященным применению AN-кодов в реальных системах, по-видимому, является исследование, проведенное Авидзинисом и др. [9] в рамках проекта JPL — STAR Лаборатории реактивных двигателей, имевшего своей целью создание сверхнадежной вычислительной машины для космических кораблей. Авидзинис и другие исследовали эффективность AN-кодов не только при исправлении ошибок, возникающих при обычных арифметических операциях, но и при сдвигах чисел, вычислении дополнительных чисел и др. Кроме того, были проведены экспериментальные исследования процессора с длиной слова 4 байта (32 бита), использовавшего AN-код с $A = 2^4 - 1$, который может быть реализован с помощью относительно простых и недорогих устройств. Результаты иссле-

[1]) Эти вопросы рассмотрены в книге Рао [12].

дования проведенного Авидзинисом и другими, в частности, доказывают высокую эффективность использования AN-кодов с порождающими числами A вида $2^b - 1$ для обнаружения ошибок в вычислительных машинах с параллельной обработкой байтов длины b.

Предполагается, что AN-коды, рассмотренные в этой и предыдущей главах, будут необходимы для достижения сверхвысокой надежности вычислительных машин, составляющих основу больших информационных систем, которые будут созданы в будущем.

Приложения

Приложение 1. Разложение чисел вида 2^n-1 на простые множители и таблица неприводимых многочленов

В табл. А.1 содержатся разложения чисел 2^n-1, $0<n\leqslant 50$, на простые множители. В этой таблице рядом с числом n стоит разложение 2^n-1 на простые множители. Отмеченные знаком * сомножители могут не быть простыми, но ни на одно простое число, меньшее 25013, они не делятся.

В табл. А.2 помещены неприводимые многочлены над $GF(p)$, $p=2, 3, 5$, в следующей записи:

$$P=p \quad M=m \quad E=e$$
$$i \qquad f_i \ \ldots$$

где f_i — минимальный многочлен элемента α^i, α — некоторый примитивный элемент. В случае $p=2$ для записи коэффициентов многочленов используется восьмеричное представление (коэффициенты двоичного представления разбиваются на группы по три символа, начиная с младших разрядов, и каждая группа записывается с помощью одного восьмеричного числа; например, двоичному представлению $\overline{011}\ \overline{101}\ \overline{001}$ соответствует восьмеричное представление 351; от двоичного представления легко перейти к восьмеричному представлению и наоборот).

Пример. В случае $p=2$, $m=4$ и $e=15$ число 23 в паре (1, 23) является восьмеричным представлением неприводимого многочлена X^4+X+1 над $GF(2)$, поскольку число 23 является восьмеричным представлением двоичной последовательности 010 011.

В случае $p=3$, $m=2$ и $e=8$ последовательность 112 в паре (1, 112) является непосредственно последовательностью коэффициентов неприводимого многочлена X^2+X+2.

Для любых p и m многочлен f_1 пары $(1, f_1)$ является примитивным многочленом, и его показатель e равен p^m-1. Если существует несколько примитивных многочленов заданной степени над $GF(p)$, для данной таблицы выбирался тот из них, который имеет минимальное число ненулевых коэффициентов, а при наличии нескольких таких многочленов — тот, последовательность коэффициентов которого, рассматриваемая как p-ичное число, является наименьшей.

Для любых p и m в таблицу помещены только неприводимые многочлены, степень которых в точности равна m. По этой

таблице можно определить минимальный многочлен любого элемента α^i, $1 \leqslant i \leqslant p^m - 2$. Если i не указано, в таблице следует найти одно из чисел i_k или i_l, таких, что $i_k = ip^k \bmod (p^m - 1)$ и $i_l = -ip^l \bmod (p^m - 1)$. Если в таблице приведено (i_k, f_{i_k}), то искомым минимальным многочленом элемента α^i является f_{i_k}. Если указана пара (i_l, f_{i_l}), то искомым минимальным многочленом элемента α^{-i} будет многочлен, двойственный к f_{i_l}. При проведении подобных вычислений удобно пользоваться таблицами циклов.

Пример. Случай $p = 3$, $m = 2$.

Из таблицы берем примитивный многочлен $X^2 + X + 2$. Найдем минимальные многочлены f_i над $GF(3)$ элементов α^i:

1) f_2: по таблице находим $(2, 101)$; следовательно, $f_2 = X^2 + 1$.

2) f_3: так как 3 в таблицу не входит, то строим таблицу циклов:

$$3, 9 (= 1);$$

следовательно, $f_3 = f_1$.

3) f_4: аналогично строим цикл

$$4, 2 (= 4);$$

так как этот цикл имеет длину 1, то искомый многочлен имеет степень 1.

4) f_5: в этом случае таблица циклов имеет вид

$$5, 15 (= 7 = -1);$$

многочленом, двойственным к f_1, является $X^2(X^{-2} + X^{-1} + 2) = 2X^2 + X + 1 = 2(X^2 - X - 1)$ (поскольку $2 = -1$); значит, $f_5 = f_7 = X^2 - X - 1$.

5) f_6: $6, 18 = 2$; следовательно, $f_6 = f_2$.

Табл. А.1 и А.2 были построены С. Адзуми. Табл. А.1 построена с помощью ЭВМ NEAC 2200-500 TSS, а табл. А.2 — с помощью ЭВМ FACOM 230-45 S.

Приложение 2. Параметры двоичных БЧХ-кодов в узком смысле длины 1023 и менее

В табл. А.3 приведены следующие параметры двоичных БЧХ-кодов в узком смысле длины 1023 и менее: n — длина кода, k — число информационных символов, $d_{\text{БЧХ}}$ — граница БЧХ

и разность $d_{\text{мин}} - d_{\text{БЧХ}}$, где $d_{\text{мин}}$ — истинное минимальное расстояние кода. Для кодов, истинное минимальное расстояние которых не определено, в графе $d_{\text{мин}} - d_{\text{БЧХ}}$ стоит знак —. Значения параметров n, k и $d_{\text{БЧХ}}$, приведенные в табл. А.3, взяты из табл. 9.1 книги Питерсона и Уэлдона. Значения $d_{\text{мин}}$ получены на основании результатов работ [9, 10, 12, 13] из списка литературы к гл. 4.

Таблица А.1

2	3	27	7.73.262657
3	7	28	3.5.29.43.113.127
4	3.5	29	233.1103.2089
5	31	30	3^2.7.11.31.151.331
6	3^2.7	31	2147483647
7	127	32	3.5.17.257.65537
8	3.5.17	33	7.23.89.599479
9	7.73	34	3.43691.131071
10	3.11.31	35	31.71.127.122921
11	23.89	36	3^3.5.7.13.19.37.73.109
12	3^2.5.7.13	37	223.616318177
13	8191	38	3.91625968981*
14	3.43.127	39	7.79.8191.121369
15	7.31.151	40	3.5^2.11.17.31.41.61681
16	3.5.17.257	41	13367.164511353
17	131071	42	$3^2.7^2$.43.127.337.5419
18	3^3.7.19.73	43	431.9719.2099863
19	524287	44	3.5.23.89.397.683.2113
20	3.5^2.11.31.41	45	7.31.73.151.631.23311
21	7^2.127.337	46	3.47.499069107643*
22	3.23.89.683	47	2351.4513.13264529
23	47.178481	48	3^2.5.7.13.17.97.241.257.673
24	$3^2.5^2$.7.13.17.241	49	127.4432676798593*
25	31.601.1801	50	3.11.31.251.601.1801.4051
26	3.2731.8191		

Таблица А.2

P=2 M=2 E=3
1 7

P=2 M=3 E=7
1 13

P=2 M=4 E=5
3 37

P=2 M=4 E=15
1 23

P=2 M=5 E=31
1 45 3 75 5 67

P=2 M=6 E=9
7 111

P=2 M=6 E=21
3 127

P=2 M=6 E=63
1 103 5 147 11 155

P=2 M=7 E=127
1 203 3 253 5 217 7 375 9 271 11 211
13 345 19 357 21 313

P=2 M=8 E=17
15 727 45 471

P=2 M=8 E=51
5 763 25 433

P=2 M=8 E=85
3 567 9 675 21 613 27 477

P=2 M=8 E=255
1 435 7 551 11 747 13 453 19 545 23 543
37 537 43 703

P=2 M=9 E=73
7 1231 21 1027 35 1401 77 1511

P=2 M=9 E=511
1 1021 3 1131 5 1461 9 1423 11 1055 13 1167
15 1541 17 1333 19 1605 23 1751 25 1743 27 1617
29 1553 37 1157 39 1715 41 1563 43 1713 45 1175
51 1725 53 1225 55 1275 75 1773 83 1425 85 1267

P=2 M=10 E=11
93 3777

P=2 M=10 E=33
31 3043 155 2251

P=2 M=10 E=93
11 2065 55 3453 77 2413

P=2 M=10 E=341
3 2017 9 2257 15 2653 21 3753 27 3573 39 2107
45 3061 51 2547 57 3121 69 2701 75 2437 87 2311
105 3607 147 2355 171 3315

P=2 M=10 E=1023
1 2011 5 2415 7 3771 13 2157 17 3515 19 2773
23 2033 25 2443 29 2461 35 3023 37 3543 41 2745
43 2431 47 3177 49 3525 53 2617 59 3471 71 3323
73 3507 83 3623 85 2707 89 2327 91 3265 101 2055
103 3575 107 3171 109 2047 149 3025 173 3337 179 3211

P=2 M=11 E=23
89 5343

Продолжение

P=2 M=11 E=89
23 4757 69 6777 115 7311 299 4303
P=2 M=11 E=2047
1 4005 3 4445 5 4215 7 4055 9 6015 11 7413
13 4143 15 4563 17 4053 19 5023 21 5623 25 4577
27 6233 29 6673 31 7237 33 7335 35 4505 37 5337
39 5263 41 5361 43 5171 45 6637 47 7173 49 5711
51 5221 53 6307 55 6211 57 5747 59 4533 61 4341
67 6711 71 7715 73 6343 75 6227 77 6263 79 5235
81 7431 83 6455 85 5247 87 5265 91 4767 93 5607
99 4603 101 6561 103 7107 105 7041 107 4251 109 5675
111 4173 113 4707 117 5463 119 5755 137 6675 139 7655
141 5531 147 7243 149 7621 151 7161 153 4731 155 4451
157 6557 163 7745 165 7317 167 5205 169 4565 171 6765
173 7535 179 4653 181 5411 183 5545 185 7565 199 6543
201 5613 203 6013 205 7647 211 6507 213 6037 215 7363
217 7201 219 7273 293 7723 301 5007 307 7555 309 4261
331 6447 333 5141 339 7461 341 5253

P=2 M=12 E=13
315 17777
P=2 M=12 E=35
117 16475
P=2 M=12 E=39
105 13617
P=2 M=12 E=45
91 10011
P=2 M=12 E=65
63 10761 189 13535 441 15353 693 12345
P=2 M=12 E=91
45 15173 135 10571 405 13003
P=2 M=12 E=105
39 13321 429 11073
P=2 M=12 E=117
35 10377 175 16757 595 13413
P=2 M=12 E=195
21 10065 147 17657 231 10167 357 15347
P=2 M=12 E=273
15 13077 75 16267 165 10621 285 17141 345 14537 435 10743
P=2 M=12 E=315
13 12513 143 12111 169 16547 221 17513 299 16401 403 14373
P=2 M=12 E=455
9 11765 27 14405 81 17545 99 17323 153 13517 171 10213
207 10603 279 16137 297 17073 333 11703 423 14313 603 14411
P=2 M=12 E=585
7 12153 49 17703 77 15115 119 14315 133 11433 203 13157
217 13563 301 12315 329 11637 343 12673 371 14043 427 10245
P=2 M=12 E=819
5 10115 25 13377 55 10201 85 11673 95 17705 115 13223
145 16535 155 14241 185 14037 205 14513 215 10653 235 11411
275 14601 295 11463 355 10467 365 11265 425 14155 685 16355
P=2 M=12 E=1365
3 12133 33 13311 51 10355 57 12331 69 16663 87 17361
93 14755 111 15415 123 13475 141 13571 177 16017 183 14111
201 14067 213 16457 219 11657 237 13737 291 15775 309 11763
327 17233 339 16003 363 11045 411 10077 597 12265 717 16571
P=2 M=12 E=4095
1 10123 11 15647 17 16533 19 16047 23 11015 29 14127
31 17673 37 13565 41 15341 43 15053 47 15621 53 15321
59 11417 61 13505 67 13275 71 11471 73 16237 79 12515
83 12255 89 11271 97 17121 101 14227 103 12117 107 14135
109 14711 113 13131 121 16521 137 15437 139 12067 149 12147

Продолжение

P=2 M=12 E=4095

151	14717	157	14675	163	10663	167	16115	173	12247	179	17675
181	10151	187	14613	197	11441	199	10321	209	11067	211	14433
227	12753	229	13431	233	11313	239	13425	277	16021	281	17025
283	15723	293	11477	307	14221	311	12705	313	14357	331	16407
341	11561	347	17711	349	13701	359	11075	361	16363	397	12727
407	17057	409	10437	421	14271	437	12623	439	12007	587	14545
589	16311	613	15413	619	17147	661	10605	683	10737	691	15701

P=2 M=13 E=8191

1	20033	3	23261	5	24623	7	23517	9	30741	11	21643
13	30171	15	21277	17	27777	19	35051	21	34723	23	34047
25	32535	27	31425	29	37505	31	36515	33	26077	35	35673
37	20635	39	33763	41	25745	43	36575	45	26653	47	21133
49	22441	51	30417	53	32517	55	37335	57	25327	59	23231
61	25511	63	26533	65	33343	67	33727	69	27271	71	25017
73	26041	75	21103	77	27263	79	24513	81	32311	83	31743
85	24037	87	30711	89	32641	91	24657	93	32437	95	20213
97	25633	99	31303	101	22525	103	34627	105	25775	107	21607
109	25363	111	27217	113	33741	115	37611	117	23077	119	21263
121	31011	123	27051	125	35477	131	34151	133	27405	135	34641
137	32445	139	36375	141	22675	143	36073	145	35121	147	36501
149	33057	151	36403	153	35567	155	23167	157	36217	159	22233
161	32333	163	24703	165	33163	167	32757	169	23761	171	24031
173	30025	175	37145	177	31327	179	27221	181	25577	183	22203
185	37437	187	27537	189	31035	195	24763	197	20245	199	20503
201	20761	203	25555	205	30357	207	33037	209	34401	211	32715
213	21447	215	27421	217	20363	219	33501	221	20425	223	32347
225	20677	227	22307	229	33441	231	33643	233	24165	235	27427
237	24601	239	36721	241	34363	243	21673	245	32167	247	21661
265	33357	267	26341	269	31653	271	37511	273	23003	275	22657
277	25035	279	23267	281	34005	283	34555	285	24205	291	26611
293	32671	295	25245	297	31407	299	33471	301	22613	303	35645
305	32371	307	34517	309	26225	311	35561	313	25663	315	24043
317	30643	323	20157	325	37151	327	24667	329	33325	331	32467
333	30667	335	22631	337	26617	339	20275	341	36625	343	20341
345	37527	347	31333	349	31071	355	23353	357	26243	359	21453
361	36015	363	36667	365	34767	367	34341	369	34547	371	35465
373	24421	375	23563	377	36037	391	31267	393	27133	395	30705
397	30465	399	35315	401	32231	403	32207	405	26101	407	22567
409	21755	411	22455	413	33705	419	37621	421	21405	423	30117
425	23021	427	21525	429	36465	431	33013	433	27531	435	24675
437	33133	439	34261	441	33405	443	34655	453	32173	455	33455
457	35165	459	22705	461	37123	463	27111	465	35455	467	31457
469	23055	471	30777	473	37653	475	24325	477	31251	547	35163
549	33433	551	37243	553	27515	555	32137	557	26743	563	30277
565	20627	567	35057	569	24315	571	24727	581	30331	583	34273
585	23207	587	31113	589	36023	595	27373	597	20737	599	36235
601	21575	603	26215	605	21211	611	20311	613	34003	615	34027
617	20065	619	22051	621	22127	627	23621	629	24465	651	26457
653	31201	659	34035	661	27227	663	22561	665	21615	667	22013
669	23365	675	26213	677	26775	679	32635	681	33631	683	32743
685	31767	691	34413	693	22037	695	30651	697	26565	711	22141
713	22471	715	35271	717	37445	723	22717	725	26505	727	24411
729	24575	731	23707	733	25173	739	21367	741	25161	743	24147
793	36307	795	24417	805	20237	807	36771	809	37327	811	27735
813	31223	819	36373	821	33121	823	32751	825	33523	839	26415
841	23737	843	25425	845	34603	851	31047	853	37305	855	21315
857	35777	859	32725	869	20571	871	30301	873	34757	875	21067
877	25151	1171	27513	1173	33721	1179	34775	1189	23571	1195	27411
1197	20457	1203	21557	1205	30177	1227	26347	1229	27477	1235	34243
1237	27235	1323	25175	1325	31231	1331	31131	1333	25503	1355	33045

Продолжение

P=2 M=13 E=8191
1357 24253 1363 35351 1365 26053

P=2 M=14 E=43
381 47771 1143 64213 2667 52225

P=2 M=14 E=129
127 51745 635 77277 889 42721 1397 41241 1651 71747 2413 62723

P=2 M=14 E=381
43 72127 215 40621 301 43353 473 67665 559 65013 731 73305
817 51161 1333 40207 1591 50133

P=2 M=14 E=5461
3 44747 9 63155 15 57521 21 63747 27 60117 33 64707
39 77741 45 62055 51 50477 57 63163 63 75505 69 42403
75 41001 81 64003 87 50357 93 65763 99 44121 105 72501
111 45257 117 76175 123 52423 135 72203 141 47755 147 70627
153 46033 159 64157 165 52547 171 63217 177 56155 183 61065
189 77631 195 67627 201 52131 207 50373 213 57761 219 42711
225 41111 231 72513 237 47043 243 54423 249 74121 261 51467
267 73655 273 62603 279 55217 285 77643 291 54021 297 74645
303 57745 309 62225 315 46605 327 55643 333 71323 339 62261
345 52553 351 77703 357 44037 363 54055 369 51547 375 63001
393 50441 399 66361 405 57653 411 66735 417 53525 423 67267
429 77001 435 45131 441 44545 453 62731 459 77661 465 71057
471 43503 477 56067 483 44367 489 46053 495 54775 531 75767
537 57143 549 74351 555 71141 561 50343 567 75167 573 64475
579 73225 585 45275 591 77211 597 54175 603 40653 615 46145
621 51371 627 43243 633 50433 651 56607 657 53513 663 60241
669 62761 675 41403 681 70443 687 77727 693 44221 699 43665
711 74117 717 43225 723 64033 729 77307 741 64311 747 70065
753 76617 783 41551 789 65373 795 56711 807 47513 813 65043
819 60523 825 47601 837 64407 843 44307 849 52737 855 71557
861 47441 867 54371 873 67541 879 40555 885 65315 909 62251
915 45437 921 65637 933 45027 939 64633 945 64215 951 61353
1095 56741 1101 51353 1107 40145 1113 66141 1125 62511 1131 54321
1173 67533 1179 46761 1191 44501 1197 44375 1203 43215 1209 73603
1221 42361 1227 51723 1239 56421 1245 61115 1251 51275 1257 40327
1305 74463 1317 66703 1323 62431 1335 71501 1353 55651 1365 46047
1371 71255 1383 50313 1389 47263 1395 41163 1431 76511 1437 72465
1443 71217 1449 72325 1461 70535 1479 64715 1485 74133 1587 53135
1593 71765 1611 77233 1623 47727 1641 70053 1683 52353 1689 63667
1701 67115 1707 77217 1719 42567 1737 44361 1749 75473 1755 70267
2349 57743 2355 75363 2379 74705 2469 54207 2475 63351 2643 40413
2733 53545 2739 53763 2763 56637

P=2 M=14 E=16383
1 40071 5 53727 7 55037 11 76715 13 47067 17 52431
19 63331 23 67737 25 70641 29 40635 31 57737 35 45145
37 43035 41 65403 47 71367 49 57017 53 45715 55 40423
59 40473 61 45553 65 61757 67 72253 71 67047 73 70665
77 76131 79 67517 83 54647 85 42631 89 47121 91 76571
95 76371 97 52621 101 64167 103 75273 107 71411 109 53477
113 63247 115 50351 119 41357 121 67207 125 63253 131 51305
133 46505 137 63073 139 50007 143 54641 145 51647 149 64635
151 45641 155 56031 157 50523 161 43017 163 73143 167 44651
169 54517 173 51633 175 67333 179 70063 181 50325 185 74313
187 67335 191 55165 193 57365 197 46107 199 41753 203 61237
205 67473 209 41015 211 40415 217 42763 221 63265 223 76621
227 60575 229 56315 233 72753 235 55427 239 66301 241 56717
245 47667 247 74661 251 77637 263 54257 265 71447 269 72717
271 56615 275 53061 277 44567 281 42163 283 64237 287 66127
289 40665 293 50765 295 55301 299 56135 305 64563 307 57475
311 72643 313 74711 317 75077 323 65453 325 43305 329 50631
331 71721 335 51213 337 62413 341 73321 343 76663 347 77015

Продолжение

P=2 M=14 E=16383

349	70035	353	41711	355	40731	359	40503	361	57541	365	71453
367	66551	371	61461	373	77525	377	73527	379	67225	389	45737
391	46517	395	46113	397	70633	401	65165	403	47111	407	70467
409	43751	413	51261	415	61565	419	66163	421	53575	425	42705
427	40137	431	61443	433	53253	437	54065	439	47551	443	50753
445	60421	451	65601	455	41177	457	77721	461	51055	463	65203
467	72375	469	72023	475	45713	479	75761	481	60411	485	47025
487	57347	491	70251	493	54353	529	65427	533	57443	535	56171
539	75053	541	71037	547	44343	551	55265	553	60025	557	57505
563	57431	565	65155	569	47357	571	40647	581	54665	583	57055
587	77411	589	64223	593	53025	595	66545	599	52571	601	47155
605	55073	611	56345	613	43131	617	52701	619	63537	623	72675
625	63227	629	41343	631	73125	647	57231	649	75033	653	62011
655	67731	659	46355	661	70425	665	45371	667	42323	677	45563
679	77747	683	63153	685	63015	689	40257	691	43745	695	55567
697	52751	701	71361	707	70553	709	53657	713	75443	715	60673
719	76707	721	54377	725	63601	727	42531	733	47203	739	42453
743	76321	745	52457	749	44141	751	74331	755	70773	781	54577
785	47351	787	45243	791	55331	793	76005	797	56557	803	70317
805	65277	809	73353	811	54617	815	65575	821	67621	823	51243
827	64653	839	51017	841	72573	845	64553	847	62401	851	64641
853	46215	857	40251	859	75653	869	70461	871	62725	875	57333
877	47233	881	71117	883	44133	887	73137	905	52215	907	43701
911	41607	913	50135	917	42013	919	45653	923	70767	925	70311
931	65755	935	51707	937	72117	941	55705	943	66075	947	61063
949	56111	953	53703	955	52027	1093	77343	1097	74557	1099	50733
1109	77561	1111	61335	1115	52655	1117	56203	1123	53623	1127	45213
1129	57407	1133	62107	1139	55105	1141	47753	1163	66661	1165	54243
1171	67425	1175	56241	1177	60657	1181	45555	1187	55417	1189	60325
1193	56625	1195	74025	1205	71337	1207	40053	1211	51321	1223	44031
1225	61533	1229	73725	1235	65133	1237	41127	1241	54627	1243	50777
1253	62677	1255	67751	1259	72721	1301	57003	1303	51303	1307	51145
1309	62467	1315	76505	1319	70475	1321	76303	1325	45065	1331	67371
1337	52137	1351	45335	1355	60147	1357	67401	1363	55543	1367	77235
1369	62767	1373	63203	1379	67011	1381	57201	1385	51677	1387	71623
1421	74601	1427	61621	1429	52237	1433	54003	1435	62671	1445	55703
1447	47103	1451	45511	1453	75617	1459	45467	1463	56463	1465	40273
1481	53057	1483	41755	1589	51347	1607	72133	1609	62473	1613	76223
1619	72631	1621	66523	1625	62655	1627	43161	1637	41625	1639	46565
1643	77477	1645	51671	1685	51005	1687	75003	1691	51231	1703	75723
1705	71061	1709	47153	1715	41657	1717	55103	1721	53233	1735	73237
1739	64457	1741	75267	1747	61363	1751	46077	1753	61333	2341	41441
2347	76627	2357	53367	2381	41205	2387	73243	2389	43517	2395	42557
2405	72351	2411	73563	2453	64305	2459	44203	2477	64035	2645	75571
2731	67451	2741	77613	2765	71615	2771	41477	2861	43173	2867	55111

P=3 M=2 E=4
2 101

P=3 M=2 E=8
1 112

P=3 M=3 E=13
2 1112

P=3 M=3 E=26
1 1021
5 1211

P=3 M=4 E=5
16 11111

Продолжение

```
P=3   M=4    E=10
   8 12121
P=3   M=4    E=16
   5 10102
P=3   M=4    E=20
   4 12011
P=3   M=4    E=40
   2 10121     14 10111
P=3   M=4    E=80
   1 10012      7 11222     11 12112     13 11002

P=3   M=5    E=11
  22 112102
P=3   M=5    E=22
  11 102221
P=3   M=5    E=121
   2 101012     4 120212     8 120022    10 112022    14 111212    16 121112
  20 102112    32 112202    34 100112    38 101102    40 110002
P=3   M=5    E=242
   1 100021     5 121111     7 102101    13 111011    17 122101    19 112001
  23 111211    31 120221    41 101011    43 110021    47 102211

P=3   M=6    E=7
 104 1111111
P=3   M=6    E=14
  52 1212121
P=3   M=6    E=28
  26 1101011 130 1202021
P=3   M=6    E=52
  14 1020001  70 1010201
P=3   M=6    E=56
  13 1022102  65 1100012
P=3   M=6    E=91
   8 1011001  16 1211201  32 1112011  40 1122001  64 1102121 128 1210211
P=3   M=6    E=104
   7 1001122  35 1121122  49 1221112 119 1221002
P=3   M=6    E=182
   4 1001221  20 1110221  44 1211021  68 1112201  92 1111021 124 1012001
P=3   M=6    E=364
   2 1001021  10 1110011  22 1011011  34 1210001  38 1202101  46 1021021
  50 1022011  58 1000111  62 1122221  94 1200121 122 1002011 142 1221211
P=3   M=6    E=728
   1 1000012   5 1211212  11 1222102  17 1202002  19 1012112  23 1222022
  25 1220212  29 1222222  31 1110202  37 1010212  41 1112222  43 1121012
  47 1102202  59 1201202  61 1120102  67 1022122  71 1101212  95 1101002
  97 1202222 101 1020112 103 1212122 113 1110122 115 1122202 121 1100002


P=5   M=2    E=3
   8 111
P=5   M=2    E=6
   4 141
P=5   M=2    E=8
   3 103
P=5   M=2    E=12
   2 134
P=5   M=2    E=24
   1 112        7 123

P=5   M=3    E=31
   4 1244       8 1404      12 1114      16 1434      32 1024
```

Продолжение

```
P=5   M=3    E=62
   2 1141      6 1201     14 1101     18 1131     38 1341
P=5   M=3    E=124
   1 1032      3 1143      7 1113      9 1302     11 1323     13 1442
  17 1042     19 1213     33 1312     37 1102

P=5   M=4    E=13
  48 13031    96 11411   192 12121
P=5   M=4    E=16
  39 10003
P=5   M=4    E=26
  24 13131    72 12021   168 14441
P=5   M=4    E=39
  16 12311    32 12201    64 10141
P=5   M=4    E=48
  13 10102    91 10203
P=5   M=4    E=52
  12 12131    36 11441    84 14011
P=5   M=4    E=78
   8 10111    56 10231    88 13341
P=5   M=4    E=104
   6 13314    18 13044    42 13444    54 14134    66 12014   162 11234
P=5   M=4    E=156
   4 11301    28 12211    44 10421    68 13241    92 13401   188 14301
P=5   M=4    E=208
   3 11213     9 13102    21 14232    27 13413    33 11402    57 13342
  63 14243    69 14402    81 14112    87 10313    93 11142   183 11133
P=5   M=4    E=312
   2 12004    14 14444    22 12224    34 13124    38 11414    46 11244
  58 12324    62 10024    82 14314    86 12434    94 11004   158 14004
P=5   M=4    E=624
   1 10122     7 11013    11 10123    17 11212    19 13023    23 13043
  29 13302    31 12203    37 11202    41 11032    43 14413    47 14143
  53 12222    59 12123    61 13032    67 12013    71 11443    79 12433
  83 13423    89 14022   157 10412   163 12033   167 10413   187 14303
```

Таблица А.3

n	k	$d_{БЧХ}$	$d_{мин}-d_{БЧХ}$	n	k	$d_{БЧХ}$	$d_{мин}-d_{БЧХ}$	n	k	$d_{БЧХ}$	$d_{мин}-d_{БЧХ}$
7	4	3	0	127	120	3	0	255	215	11	0
					113	5	0		207	13	0
15	11	3	0		106	7	0		199	15	0
	7	5	0		99	9	0		191	17	0
	5	7	0		92	11	0		187	19	0
31	26	3	0		85	13	0		179	21	—
	21	5	0		78	15	0		171	23	0
	16	7	0		71	19	0		163	25	0
	11	11	0		64	21	0		155	27	0
	6	15	0		57	23	0		147	29	—
					50	27	0		139	31	0
63	57	3	0		43	29	2		131	37	—
	51	5	0		36	31	0		123	39	—
	45	7	0		29	43	—		115	43	—
	39	9	0		22	47	0		107	45	0
	36	11	0		15	53	0		99	47	0
	30	13	0		8	63	0		91	51	0
	24	15	0						87	53	—
	18	21	0	255	247	3	0		79	55	0
	16	23	0		239	5	0		71	59	—
	10	27	0		231	7	0		63	61	—
	7	31	0		223	9	0		55	63	0

Продолжение

n	k	$d_{БЧХ}$	$d_{мин}-d_{БЧХ}$	n	k	$d_{БЧХ}$	$d_{мин}-d_{БЧХ}$	n	k	$d_{БЧХ}$	$d_{мин}-d_{БЧХ}$
255	47	85	0	511	67	175	—	1023	543	107	—
	45	87	—		58	183	—		533	109	—
	37	91	—		49	187	—		523	111	0
	29	95	0		40	191	0		513	115	—
	21	111	0		31	219	0		503	117	—
	13	119	0		28	223	0		493	119	0
	9	127	0		19	239	0		483	121	0
					10	255	0		473	123	—
511	502	3	0						463	125	—
	493	5	0	1023	1013	3	0		453	127	0
	484	7	0		1003	5	0		443	147	—
	475	9	0		993	7	0		433	149	—
	466	11	0		983	9	0		423	151	—
	457	13	0		973	11	0		413	155	—
	448	15	0		963	13	0		403	157	—
	439	17	—		953	15	0		393	159	—
	430	19	0		943	17	—		383	165	0
	421	21	—		933	19	0		378	167	—
	412	23	0		923	21	0		368	171	—
	403	25	—		913	23	0		358	173	—
	394	27	0		903	25	—		348	175	—
	385	29	—		893	27	—		338	179	—
	376	31	0		883	29	—		328	181	—
	367	33	—		873	31	0		318	183	—
	358	37	—		863	33	0		308	187	—
	349	39	—		858	35	—		298	189	—
	340	41	—		848	37	—		288	191	0
	331	43	—		838	39	—		278	205	—
	322	45	—		828	41	—		268	207	—
	313	47	0		818	43	—		258	213	—
	304	51	—		808	45	0		248	215	—
	295	53	—		798	47	0		238	219	—
	286	55	—		788	49	—		228	221	—
	277	57	—		778	51	—		218	223	0
	268	59	0		768	53	—		208	231	0
	259	61	—		758	55	0		203	235	—
	250	63	—		748	57	—		193	237	—
	241	73	—		738	59	—		183	239	0
	238	75	0		728	61	—		173	245	—
	229	77	—		718	63	0		163	247	—
	220	79	—		708	69	—		153	251	—
	211	83	—		698	71	—		143	253	—
	202	85	—		688	73	—		133	255	0
	193	87	—		678	75	—		123	341	—
	184	91	—		668	77	0		121	343	—
	175	93	2		658	79	—		111	347	—
	166	95	0		648	83	—		101	351	—
	157	103	—		638	85	—		91	363	0
	148	107	—		628	87	—		86	367	—
	139	109	2		618	89	—		76	375	—
	130	111	0		608	91	—		66	379	4
	121	112	2		598	93	0		56	383	0
	112	119	0		588	95	0		46	439	—
	103	123	—		578	99	0		36	447	0
	94	125	2		573	101	—		26	479	0
	85	127	0		563	103	—		16	495	0
	76	171	—		553	105	—		11	511	0

Литература [1])

К главе 1

1. Киясу, Мурога, Теория информации, Серия «Современная прикладная математика», изд-во «Иванами», 1957.
2. Peterson W. W., Weldon E. J., Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972; есть русский перевод: Питерсон У., Уэлдон Э., Коды, исправляющие ошибки, изд-во «Мир», 1976.
3. Hamming R. W., Error Detecting and Error Correcting Codes, *BSTJ*, **29**, 147—160 (1950); есть русский перевод: Хэмминг Р. В., Коды с обнаружением и исправлением ошибок, в сб. *Коды с обнаружением и исправлением ошибок*, ИЛ, 1956, стр. 7—22.
4. Berlekamp E. R., Algebraic Coding Theory, McGraw-Hill, N. Y., 1968; есть русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, изд-во «Мир», 1971.
5. Hsiao M. Y., A Class of Optimal Minimum Odd-Weigth-Column SEC-DED Codes, *IBM J. Res. Develop.*, **14**, 395—401 (1970).
6. Lee C. Y., Some Properties of Nonbinary Error-Correcting Codes, *IEEE Trans.*, **IT-4**, 77—82 (1958).
7. Forney G. D., Jr., Concatenated Codes, MIT Press, Cambridge, Mass., 1966; есть русский перевод: Форни Г. Д., Каскадные коды, изд-во «Мир», 1970.
8. Weldon E. J., Jr., Decoding Binary Block Codes on Q-ary Output Channels, *IEEE Trans.*, **IT-17**, 713—718 (1971).
9. Shannon C. E., Weaver W., Mathematical Theory of Communication, Univ. of Illinois Press, Urbana, Illinois, 1949; есть русский перевод: Шеннон К., Работы по теории информации и кибернетике, ИЛ, 1962, стр. 7—32.
10. Callager R. G., Information Theory and Reliable Communication, Wiley, N. Y., 1968; есть русский перевод: Галлагер Р., Теория информации и надежная связь, изд-во «Советское радио», 1974.
11. Cocke J., Lossless Symbol Coding with Nonprimes, *IRE Trans.*, **IT-5**, 33—34 (1959).
12. Васильев Ю. Л., О негрупповых плотно упакованных кодах, *Проблемы кибернетики*, № 8, 337—339 (1962).
13. Schönheim J., On Linear and Nonlinear Single-Error-Correcting q-nary Perfect Codes, *Inform. and Control*, **12**, 23—26 (1968).
14. Golay M. J. E., Notes on Digital Coding, *Proc. IRE*, **37**, 657 (1949).
15. Tietäväinen A., On the Nonexistence of Perfect Codes over Finite Fields, *SIAM J. Appl. Math.*, **24**, 88—96 (1973).
16. Plotkin M., Binary Codes with Specified Minimum Distance, *IRE Trans.*, **IT-6**, 445—450 (1960); есть русский перевод: Плоткин М., Двоичные коды с заданным минимальным расстоянием, *Кибернетический сборник*, вып. 7, ИЛ, 1963, стр. 60—73.
17. Freiman C. V., Optimal Error Detection Codes for Completely Asymmetric Binary Channels, *Inform. and Control*, **5**, 64—71 (1962).
18. Tokura N., Kasami T., Hasimoto A., Failsafe Logic Nets, *IEEE Trans.*, **C-20**, 323—330 (1971).
19. Икэно, Тамура, Намура, Равновесные коды и блок-схемы, *Дэнси цусин гаккай ромбунси*, **54-A**, 12, 671—678 (1971).

---

[1]) Литература, отмеченная звездочкой, добавлена переводчиком. — *Прим. ред.*

20. Gallager R. G., A Simple Derivation of the Coding Theorem and Some Applications, *IEEE Trans.*, **IT-11**, 3—18 (1965); есть русский перевод: Галлагер Р., Простой вывод теоремы кодирования и некоторые применения, Кибернетический сборник, новая серия, вып. 3, изд-во «Мир», 1966.
21. Shannon C. E., Gallager R. G., Berlekamp E. R., Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels, *Inform. and Control*, **10**, 65—103 (Part I), 522—552 (Part II) (1967); есть русский перевод: Шеннон К., Галлагер Р. Г., Берлекэмп Е. Р., Нижние границы вероятности ошибки кодирования в дискретных каналах без памяти, *Зарубежная радиоэлектроника*, **2**, 41—64 (1968).
22. Saltzberg B. R., Intersymbol Interference Error Bounds with Applications to Ideal Bandlimited Signaling, *IEEE Trans.*, **IT-14**, 563—568 (1968).
23. Yeh Y. S., Ho E. Y., Improved Intersymbol Interference Error Bounds in Digital Systems, *BSTJ*, **50**, 2585—2598 (1971).
24. Prabhu U. K., Some Considerations of Error Bounds in Digital Systems, *BSTJ*, **50**, 3127—3151 (1971).
25. Glave F. E., An Upper Bound on the Probability of Error due to Intersymbol Interference for Correlated Digital Signals, *IEEE Trans.*, **IT-18**, 356—363 (1972).
26*. Бассалыго Л. А., Зиновьев В. А., Леонтьев В. К., Фельдман Н. И., Несуществование совершенных кодов для некоторых составных алфавитов, *Проблемы передачи информации*, **11**, № 3, 3—13 (1975).
27*. Бассалыго Л. А., Новые верхние границы для кодов, исправляющих ошибки, *Проблемы передачи информации*, **1**, № 4, 41—44 (1965).

К главе 2

1. Peterson W. W., Weldon E. J., J., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972; есть русский перевод: Питерсон У., Уэлдон Э., Коды, исправляющие ошибки, изд-во «Мир», 1976.
2. Berlekamp E. R., Algebraic Coding Theory, McGraw-Hill, N. Y., 1968; есть русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, изд-во «Мир», 1971.
3. Янаги, Кодаира, Введение в современную математику, I, изд-во «Иванами», 1961.
4. Виноградов И. М., Основы теории чисел, изд-во «Наука», 1965.
5. Carmichael R. D., Introduction to the Theory of Groups of Finite Order, Dover, N. Y., 1956.
6. Масияма, Планирование экспериментов, Второе издание, изд-во «Иванами».
7. Hall M., Jr., Combinatorial Theory, Blaisdell Pub. Co., 1967; есть русский перевод: Холл М., Комбинаторика, изд-во «Мир», 1970.
8. Киясу, Матрицы Адамара и их применение, *Дэнси цусин гаккай ромбунси*, **57**, 17—27 (1974).

К главе 3

1. Киясу, Мурога, Теория информации, Серия «Современная прикладная математика», изд-во «Иванами», 1957.
2. Slepian D., A Class of Binary Signaling Alphabets, *BSTJ*, **35**, 203—234 (1956); есть русский перевод: Слепян Д., Класс двоичных сигнальных алфавитов, в сб. «Теория передачи сообщений» ИЛ, 1957, стр. 82—113.
3. Nordstrom A. W., Robinson J. P., An Optimum Nonlinear Code, *Inform. and Control*, **11**, 613—616 (1967).
4. Preparata E. P., A Class of Optimum Nonlinear Double Error Correcting Codes. *Inform. and Control*, **13**, 378—400 (1968); есть русский перевод: Препарата Ф. П., Класс оптимальных нелинейных кодов с исправлением

двойных ошибок, Кибернетический сборник, новая серия, вып. 7, изд-во «Мир», 1970, стр. 18—42.
5. Миякава, Имаи, Накадзима, Модифицированные коды Препарата как систематические оптимальные нелинейные коды, исправляющие двойные ошибки, *Дэнси цусин гаккай ромбунси*, **53-A**, 10, 531—539 (1970).
6. Sloane N. J. A., Reddy S. M. Chen C. L., New Binary Codes, *IEEE Trans.*, **IT-18**, 503—510 (1972).
7. Kerdock A. M., A Class of Low-Rate Nonlinear Binary Codes, *Inform. and Control*, **20**, 182—187 (1972); есть русский перевод: Кердок А. М., Класс нелинейных двоичных кодов с низкой скоростью передачи, Кибернетический сборник, новая серия, вып. 10, 1973.
8. Peterson W. W., Weldon E. J., Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972; есть русский перевод: Питерсон У., Уэлдон Э., Коды, исправляющие ошибки, изд-во «Мир», 1976.
9. Berlekamp E. R., Algebraic Coding Theory, McGraw-Hill, N. Y., 1968; есть русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, изд-во «Мир», 1971.
10. Wagner T. J., A Remark Concerning the Existence of Binary Quasi-Perfect Codes, *IEEE Trans.*, **IT-12**, 401 (1966).
11. Gorenstein D., Peterson W. W., Zierler N., Two Error-Correcting Bose — Chandhuri Codes are Quasi Perfect, *Inform. and Control*, **3**, 291—294 (1960); есть русский перевод: Горенстейн Д., Питерсон У., Цирлер Н., Квазисовершенность кодов Боуза — Чоудхури с исправлением двух ошибок, Кибернетический сборник, вып. 6, ИЛ, 1963, стр. 20—24.
12. Reiger S. H., Codes for the Correction of Clustered Errors, *IRE Trans.*, **IT-6**, 16—21 (1960).
13. Tokura N., Taniguchi K., Kasami T., A Search Procedure for Finding Optimum Group Codes for the Binary Symmetric Channel, *IEEE Trans.*, **IT-13**, 587—595 (1967).
14. Gilbert E. N., A Comparison of Signalling Alphabets, *BSTJ*, **31**, 504—522 (1952).
15. Варшамов Р. Р., Оценка числа сигналов в кодах с коррекцией ошибок, *ДАН СССР*, **117**, № 5, 739—741 (1957).
16. Sacks G. E., Multiple Error Correction by Means of Parity Checks, *IRE Trans.*, **IT-4**, 145—147 (1958).
17. MacWilliams E. J., A Theorem on the Distribution of Weights in a Systematic Code, *BSTJ*, **42**, 79—94 (1963).
18. Assmus E. F., Mattson H. F., Turyn R., Cyclic Codes, Scientific Report, AFCRL-65-322, Air Force Cambridge Research Labs., Bedford, Mass., 1965.
19. Forney G. D., Jr., Concatenated Codes, MIT Press, Cambridge, Mass., 1966; есть русский перевод: Форни Г. Д., Каскадные коды, изд-во «Мир», 1970.
20. Kasami T., Lin S., Peterson W. W., Some Results on Weight Distributions of BCH Codes, *IEEE Trans.*, **IT-12**, 274 (1966).
21. Kasami T., Weight Distribution of BCH Codes, Ch. 20 in: Combinatorial Mathematics and its Applications, Bose R. C., Dowling T. A., eds., The Univ. of North Carolina Press, Chapel Hill, N. C., 1969.
22. Sloane N. J. A., Berlekamp E. R., Weight Enumerator for Second-Order Reed-Muller Codes, *IEEE Trans.*, **IT-16**, 745—751 (1970).
23. Kasami T., Tokura N., On the Weight Structure of Reed-Muller Codes, *IEEE Trans.*, **IT-16**, 752—759 (1970).
24. Berlekamp E. R., The Weight Enumerators for Certain Subcodes of the 2nd Order Binary Reed-Muller Code, *Inform. and Control*, **17**, 485—500 (1970).
25. Kasami T., The Weight Enumerators for Several Classes of Subcodes of the 2nd Order Binary Reed-Muller Codes, *Inform. and Control*, **18**, 369—394 (1971).

26. Baumert L. D., McEliece R. J., Weights of Irreducible Cyclic Codes, *Inform. and Control*, **20**, 158—175 (1972).
27. Pless V., Powler Moment Identities on Weight Distributions in Error Correcting Codes, *Inform. and Control*, **6**, 147—152 (1963).
28. MacWilliams F. J., Sloane N. J. A., Goethals J. M., The MacWilliams Identities for Nonlinear Codes, *BSTJ*, **51**, 803—819 (1972).
29. Prange E., Cyclic Error-Correcting Codes in Two Symbols, AFCRC-TN-57-103, Air Force Cambridge Research Center, Cambridge Mass., 1957.
30. Peterson W. W., Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes, *IRE Trans.*, **IT-6**, 459—470 (1960).
31. Касами Т., О систематических кодах на основе двоичных последовательностей регистров сдвига, *Дзёхо сёри*, **1**, 198—205 (1960).
32. Rudolph L. D., Easily Implemented Error-Correction Encoding-Decoding, G. E. Tech. Rept., 62 MCD 2, Oklahoma City, Okla., 1962.
33. Prange E., The Use of Information Sets in Decoding Cyclic Codes, *IRE Trans.*, **IT-8**, S5—S9 (1962).
34. Kasami T., A Decoding Procedure for Multiple Error-Correcting Cyclic Codes, *IEEE Trans.*, **IT-10**, 134—139 (1964).
35. Rudolf L. D., Mitchell M. E., Implementation of Decoders for Cyclic Codes, *IEEE Trans.*, **IT-10**, 259—260 (1964).
36. Lin S., Introduction to Error to Correcting Codes, Prentice-Hall, Englewood Cliffs, N. J., 1970.
37. MacWilliams F. J., Permutation Decoding of Systematic Codes, *BSTJ*, **43**, 485—505 (1964).
38. Kasami T., An Upper Bound on k/n for Affine-Invariant Codes with Fixed d/n, *IEEE Trans*, **IT-15**, 174—176 (1969).
39*. Блох Э. Л., Зяблов В. В., Обобщенные каскадные коды, изд-во «Связь», 1976.
40*. Бассалыго Л. А., Формализация задачи о сложности задания кода, *Проблемы передачи информации*, **12**, № 4, 105—106 (1976).

К главе 4

1. Hocquenghem A., Codes correcteurs d'erreures, Chiffres 2, 147—156 (1959).
2. Bose R. C., Ray-Chandhuri D. K., On a Class of Error Correcting Codes, *Inform. and Control*, **3**, 68—79 (1960); есть русский перевод: Боуз Р. К., Рой-Чоудхури Д. К., Об одном классе двоичных групповых кодов с исправлением ошибок, Кибернетический сборник, вып. 2, ИЛ, 1961, стр. 83—94.
3. Berlekamp E. R., Algebraic Coding Theory, McGraw-Hill, N. Y., 1968; есть русский перевод: Берлекэмп Э., Алгебраическая теория кодирования, изд-во «Мир», 1971.
4. Peterson W. W., Weldon E. J., Jr., Error-Correcting Codes, 2nd ed., MIT Press, Cambridge, Mass., 1972; есть русский перевод: Питерсон У., Уэлдон Э., Коды, исправляющие ошибки, изд-во «Мир», 1976.
5. Коя, Матрицы и определители, изд-во «Байфукан», 1957.
6. Bussey W. H., Galois Field Tables of Orderless than 100, *Bull. Amer. Math. Soc.*, **16**, 188—206 (1909).
7. Prange E., Cyclic Error-Correcting Codes in two Symbols, AFCRC-TN-57, 103, Air Force Cambridge Research Center, Cambridge, Mass., 1957.
8. Reed I. S., Solomon G., Polynomial Codes over Certain Finite Fields, *SIAM J. Appl. Math.*, **8**, 300—304 (1960); есть русский перевод: Рид И. С., Соломон Г., Полиномиальные коды над некоторыми полями, Кибернетический сборник, вып. 7, ИЛ, 1963, стр. 74—79.
9. Kasami T., Tokura N., Some Remarks on BCH Bounds and Minimum Weights of Binary Primitive BCH Codes, *IEEE Trans.*, **IT-15**, 408—413 (1969).

10. Касами Т., Лин С., Питерсон У., О свойствах линейных кодов, инвариантных относительно аффинных преобразований, и минимальное расстояние БЧХ-кодов, *Дэнси цусин гаккай ромбунси*, **50**, 1617—1622 (1972).
11. Chen C. L., Lin S., Further Results on Polynomial Codes, *Inform. and Control*, **15**, 38—60 (1969).
12. Lin S., Weldon E. J., Jr., Further Results on Cyclic Product Codes, *IEEE Trans.*, **IT-16**, 445—451 (1970).
13. Kasami T., Lin S., Some Results on the Minimum Weight of Primitive BCH Codes, *IEEE Trans.*, **IT-18**, 824—825 (1972).
14. Hartman C. R. P., Tzeng K. K., Chien R. T., Some Results on the Minimum Distance Structure of Cyclic Codes, *IEEE Trans.*, **IT-18**, 402—408 (1972).
15. Peterson W. W., Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes, *IRE Trans.*, **IT-6**, 459—470 (1960).
16. Gorenstein D., Zierler N., A Class of Cyclic Linear Codes in p^m Symbol, *SIAM J. Appl. Math.*, **9**, 207—214 (1961); есть русский перевод: Горенстейн Д., Цирлер Н., Класс кодов из p^m символов с исправлением ошибок, Кибернетический сборник, вып. 7, ИЛ, 1963.
17. Messey J. L., Shift Register Synthesis and BCH Decoding, *IEEE Trans.*, **IT-15**, 122—127 (1969).
18. Forney G. D., Jr., On Decoding BCH Codes, *IEEE Trans.*, **IT-11**, 549—557 (1965).
19. Trench W. F., An Algorithm for the Inversion of Finite Teoplitz Matrices, *J. Soc. Indust. Appl. Math.*, **12**, 515—522 (1964).
20. Ван дер Варден Б. Л., Алгебра, изд-во «Наука», 1976 (перевод с немецкого).
21. Hartmann C. R. P., Tzeng K. K., Generalization of the BCH Bound, *Inform. and Control*, **20**, 489—498 (1972).
22. Massey J. L., Threshold Decoding, MIT Press Research Monograph 20, Cambridge, Mass., 1963.
23. Lin S., Introduction to Error Correcting Codes, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1970.
24. Weldon E. J., Jr., Some Results on Majority-Logic Decoding, Ch. 8 in: Error Correcting Codes, Mann N., ed., Wiley, 1968.
25. Reed I. S., A Class of Multiple-Error-Correcting Codes and the Decoding Sheme, *IRE Trans.*, PGIT-4, 38—49 (1954); есть русский перевод: Рид И. С., Класс кодов с исправлением ошибок и схема декодирования, Кибернетический сборник, вып. 1, ИЛ, 1960, стр. 189—205.
26. Rudolf L. D., A Class of Majority-Logic Decodable Codes, *IEEE Trans.*, **IT-13**, 305—307 (1967).
27. Rudolf L. D., Threshold Decoding of Cyclic Codes, *IEEE Trans.*, **IT-15**, 414—418 (1969).
28. Gore W. C., Generalized Threshold Decoding of Linear Codes, *IEEE Trans.*, **IT-15**, 590—592 (1969).
29. Chow D., On threshold Decoding of Cyclic Codes, *Inform. and Control*, **13**, 471—483 (1968).
30. Chen C. L., On Majority-Logic Decoding of Finite Geometry Codes, *IEEE Trans.*, **IT-17**, 332—336 (1971).
31. Kasami T., Lin S., On the Construction of a Class of Majority Logic Decodable Codes, *IEEE Trans.*, **IT-17**, 600—610 (1971).
32. Mattson H. F., Solomon G., A New Treatment of Bose-Chandhuri Codes, *SIAM J. Appl. Math.*, **9**, 654—669 (1961).
33. Митани, О кодах, обнаруживающих и исправляющих ошибки, *Дэнки сикэндзё ихо*, **15**, 5, 18—22 (1951).
34. Muller D. E., Application of Boolean Algebra to Switching Circuit Design and to Error Detection, *IRE Trans.*, **EC-3**, 6—12 (1954).
35. Kasami T., Lin S., Peterson W. W., New Generalizations of the Reed-Muller Codes, Part I: Primitive Codes, *IEEE Trans.*, **IT-14**, 189—199 (1968).

36. Касами Т., Лин С., Питерсон У., О кодах Рида — Малера и их обобщении, *Дэнси цусин гаккай ромбунси*, **51-C**, 98—105 (1968).
37. Kasami T., Lin S., Peterson W. W., Polynomial Codes, *IEEE Trans.*, **IT-40**, 807—814 (1968).
38. Weldon E. J., Jr., Euclidean Geometry Cyclic Codes, in: Proc. Symp. Combinatorial Math., Univ. North Carolina, Chapel Hill, 1967.
39. Weldon E. J., Jr., New Generalizations of the Reed — Muller Codes, Part II: Nonprimitive Codes, *IEEE Trans.*, **IT-14**, 199—206 (1968).
40. Goethals J. M., Delsarte P., On a Class of Majority-Logic Decodable Cyclic Codes, *IEEE Trans.*, **IT-14**, 182—188 (1968); есть русский перевод: Гётале Дж. М., Делзарт П., Один класс циклических кодов с мажоритарным декодированием, Кибернетический сборник, новая серия, вып. 6, изд-во «Мир», 1969.
41. Lin S., On a Class of Cyclic Codes, in: Error-Correcting Codes, Mann H. B., ed., 1968, pp. 131—148.
42. Delsarte P., On Cyclic Codes that are Invariant under the General Linear Group, *IEEE Trans.*, **IT-16**, 760—769 (1970).
43. Kasami T., Lin S., On Majority-Logic Decoding for Duals of Primitive Polynomial Codes, *IEEE Trans.*, **IT-17**, 322—331 (1971).
44. Weldon E. J., Jr., Difference-Set Cyclic Codes, *BSTJ*, **45**, 1045—1055 (1966); есть русский перевод: Велдон Ю. Дж., Циклические коды, задаваемые разностными множествами, в сб. *Некоторые вопросы теории кодирования*, изд-во «Мир», 1970, стр. 9—21.
45. Lin S., Multifold Euclidean Geometry Codes, *IEEE Trans.*, **IT-19**, 537—548 (1973).
46. Delsarte P., A Geometric Approach to a Class of Cyclic Codes, *J. Comb. Th.*, **6**, 340—359 (1969).
47. Lin S., On the Number of Information Symbols in Polynomial Codes, *IEEE Trans.*, **IT-18**, 785—794 (1972).
48. Hamada N., The Rank of the Incident Matrix of Points of d-Flats in Finite Geometries, *J. Sci. Hiroshima Univ.*, **32**, 381—396 (1968).
49. Forney G. D., Jr., Concatenated Code, MIT Press Research Monograph, 37, 1966; есть русский перевод: Форни Г. Д., Каскадные коды, изд-во «Мир», 1970.
50. Justesen J., A Class of Constructive Asymptotically Good Algebraic Codes, *IEEE Trans.*, **IT-18**, 652—656 (1972).
51. Berlekamp E. R., Justesen J., Some Long Binary Codes are not so Bad, *IEEE Trans.*, **IT-20**, 351—356 (1974).
52. Berlekamp E. R., Long Primitive Binary BCH Codes Have Distance $d \sim 2n \ln R^{-1}/\log n$, *IEEE Trans.*, **IT-18**, 415—426 (1972).
53. Гоппа В. Д., Новый класс линейных корректирующих кодов, *Проблемы передачи информации*, **6**, вып. 3, 24—30 (1970).
54. Гоппа В. Д., Рациональное представление кодов и (L, g)-коды, *Проблемы передачи информации*, **7**, вып. 3, 41—49 (1971).
55. Berlekamp E. R., Goppa Codes, *IEEE Trans.*, **IT-19**, 590—592 (1973).
56. Berlekamp E. R., Moreno O., Extended Double-Error-Correcting Binary Goppa Codes are Cyclic, *IEEE Trans.*, **IT-19**, 817—818 (1973).
57. Reiger S. N., Codes for the Correction of Clustered Errors, *IRE Trans.*, **IT-6**, 16—21 (1960).
58. Fire P., A Class of Mutiple-Error-Correcting Binary Codes for Nonindepent Errors, Sylvania Report RSL-E-2, Sylvania Electronic Defense Lab., Moutain View, California, 1959.
59. Melas C. M., Gorog E., A Note on Extending Certain Codes to Correct Error Bursts in Longer Messages, *IBM J. Res. Develop.*, **7**, 151—152 (1963).
60. Abramson N. M., A Class of Systematic Codes for Non-Independent Errors, *IRE Trans.*, **IT-5**, 150—157 (1959).

61. Касами Т., Систематические циклические коды, исправляющие группирующиеся ошибки, *Дэнси цусин гаккай ромбунси*, **45**, 9—16 (1962).
62. Elspas B., Short R. A., A Note on Optimum Burst-Error-Correcting Codes, *IRE Trans.*, **IT-8**, 39—42 (1962).
63. Касами Т., Построение с помощью вычислительной машины кодов, исправляющих группирующиеся ошибки, *Дэнси цусин гаккай ромбунси*, **45**, 1527—1532 (1962).
64. Kasami T., Matoba S., Some Efficient Shortened Cyclic Codes for Burst-Error-Correction, *IEEE Trans.*, **IT-10**, 252—253 (1964).
65. Lucky R. W., Salz J., Weldon E. J., Jr., Principles of Data Communication, McGraw-Hill, N. Y., 1968.
66. Hsu H. T., Kasami T., Chien R. T., Error-Correcting Codes for a Compound Channel, *IEEE Trans.*, **IT-14**, 135—139 (1968).
67. Stone J. J., Multiple Burst Error Correction, *Inform. and Control*, **4**, 324—331 (1961).
68. Имаи, Двумерные коды, исправляющие пачки ошибок, *Дэнси цусин гаккай ромбунси*, **55-A**, 385—392 (1972).
69. Kasami T., Construction and Decomposition of Cyclic Codes of Composite Length, *IEEE Trans.*, **IT-20**, 680—683 (1974).
70. Сугияма, Такахара, Хирасава, Намекава, О кодах Гоппы, Научное общество электроники и связи, Материалы исследований по распознаванию образов, PRL 73-77, январь 1974.
71. Гоппа В. Д., Частное сообщение.
72*. Блох Э. Л., Зяблов В. В., Обобщенные каскадные коды, изд-во «Связь», 1976.

К главе 5

1. Elias P., Coding for Noisy Channel, IRE Convention Record, Part 4, 37—46 (1955).
2. Massey J. T., Threshold Decoding, MIT Press, Cambridge Mass., 1963; есть русский перевод: Месси Дж., Пороговое декодирование, изд-во «Мир», 1966.
3. Robinson J. P., Bernstein A. J., A Class of Binary Recurrent Codes with Limited Error Propagation, *IEEE Trans.*, **IT-13**, 106—113 (1967).
4. Hagelbarger D. W., Recurrent Codes: Easily Machinized, Burst-Correcting, Binary Codes, *BSTJ*, **38**, 969—984 (1959).
5. Berlekamp E. R., Note on Recurrent Codes, *IEEE Trans.*, **IT-10**, 257—258 (1964).
6. Wyner A. D., Ash R. B., Analysis of Recurrent Codes, *IEEE Trans.*, **IT-9**, 143—156 (1963).
7. Preparata F. P., Systematic Constructions of Optimal Linear Recurrent Codes for Burst-Error-Correction, *Calcolo*, **2**, 1—7 (1964).
8. Massey J. L., Inplementation of Burst-Correcting Convolutional Codes, *IEEE Trans.*, **IT-11**, 416—422 (1965).
9. Kohlenburg A., Random and Burst-Error-Correction, 1st IEEE Annual Communications Convention, Boulder, Colo., June 7—9, 1965.
10. Ивадари, Об оптимальных рекуррентных кодах, исправляющих пачки ошибок. Научное общество электросвязи, материалы исследований по теории информации, октябрь 1966.
11. Iwadare Y., On Type B-1 Burst-Error-Correcting Convolutional Codes, *IEEE Trans.*, **IT-14**, 577—583 (1968).
12. Iwadare Y., Simple and Efficient Procedures of Burst-Error Correction, Dissertation for the Degree of Dr. of Engineering, Univ. of Tokyo, 1967.
13. Tong S. Y., Systematic Construction of Self-Orthogonal Diffuse Codes, *IEE Trans.*, **IT-16**, 594—604 (1970).

14. Wozencraft J. M., Sequential Decoding for Reliable Communication, MIT Research Lab. of Electronics Tech. Report 325, Cambridge, Mass., 1957.
15. Perry K. M., Wozencraft J. M., Seco: a Self-Regulating Error-Correcting Coder-Decoder, *IRE Trans.*, **IT-8**, 128—135 (1962).
16. Wozencraft J. M., Reiffen B., Sequential Decoding, MIT Press, Cambridge, Mass., 1961; есть русский перевод: Возенкрафт Дж., Рейффен Б., Последовательное декодирование, ИЛ, 1963.
17. Fano R. M., A Heuristic Discussion of Probabilistic Decoding, *IEEE Trans.*, **IT-9**, 64—73 (1963); есть русский перевод: Фано Р., Эвристическое обсуждение вероятностного декодирования, в сб. *Теория кодирования*, изд-во «Мир», 1964, стр. 166—198.
18. Savage J. E., The Computation Problem with Sequential Decoding, MIT Research Lab. of Electronics, Tech. Rep. 439, 1965.
19. Savage J. E., Progress in Sequential Decoding, in: Advances in Communication Systems, Balakrishnan A. V., ed., Academic Press, N. Y., 1968.
20. Jacobs I. E., Berlekamp E. R., A Lower Bound to the Distribution of Computation for Sequential Decoding, *IEEE Trans.*, **IT-13**, 167—174 (1967).
21. Jordan K. L., Jr., The Performance of Sequential Decoding in Conjunction with Efficient Modulation, *IEEE Trans.*, **COM-14**, 283—296 (1966).
22. Bussgang J. J., Some Properties of Binary Convolutional Code Generators, *IEEE Trans.*, **IT-11**, 90—99 (1965).
23. Lin S., Lyne H., Some Results of Binary Convolutional Code Generators, *IEEE Trans.*, **IT-13**, 134—139 (1967).
24. Costello D. J., Jr., A Construction Technique for Random-Error-Correcting Convolutional Codes, *IEEE Trans.*, **IT-15**, 631—636 (1969).
25. Viterbi A. J., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, *IEEE Trans.*, **IT-13**, 260—269 (1967); есть русский перевод: Витерби А., Границы ошибок для сверточных кодов и асимптотически оптимальный алгоритм декодирования, в сб. *Некоторые вопросы теории кодирования*, изд-во «Мир», 1970, стр. 142—165.
26. Jelinek F., Fast Sequential Decoding Algorithm Using a Stack, *IBM J. Res. Develop.*, **13**, 675—685 (1969).
27. Зигангиров К. Ш., Некоторые последовательные процедуры декодирования, *Проблемы передачи информации*, **2**, 13—25 (1966).
28. Yudkin H. L., Channel State Testing in Information Decoding, Sc. D. Thesis, Dept. of Elec. Engrg., MIT, Cambridge, Mass., 1964.
29. Forney G. D., Jr., Convolutional Codes I: Algebraic Structure, *IEEE Trans.*, **IT-16**, 720—738 (1970).
30. Massey J. L., Application of Automata Theory in Coding in: Applied Automata Theory, Ton J. T., ed., Academic Press, N. Y., 1968.
31. Huffman D. A., The Synthesis of Linear Sequential Coding Network, in: Information Theory, Cherry C., ed., Academic Press, N. Y., 1956.
32. Massey J. L., Liu R. W., Application of Lyapunov's Direct Method to the Error-Propagation Effect in Convolutional Codes, *IEEE Trans.*, **IT-10**, 248—250 (1964).
33. Robinson J. P., Error Propagation and Definite Decoding of Convolutional Codes, *IEEE Trans.*, **IT-14**, 121—128 (1968).
34. Massey J. L., Неопубликованный меморандум, July 1967.
35. Sullivan D. D., Error Propagation Properties of Uniform Codes, *IEEE Trans.*, **IT-15**, 152—161 (1969).
36. Robinson J. P., Punctured Uniform Codes, *IEEE Trans.*, **IT-15**, 149—152 (1969).
37. Singer J., A. Theorem in Finite Projective Geometry and Some Applications to Number Theory, AMS Trans., **43**, 377—385 (1938).
38. Falconer D. D., A Hybrid Coding Scheme for Discrete Memoryless Channels, *BSTJ*, **47**, 691—728 (1968).

39. Lucky R. W. et al., Principles of Data Communication, McGraw-Hill, N. Y., 1968.
40. Townsend R. L., Weldon E. J., Jr., Self-Orthogonal Quasi-Cyclic Codes, *IEEE Trans.*, **IT-13**, 183—195 (1967).
41. Gallager R. G., Information Theory and Reliable Communication, Wiley, N. Y., 1968; есть русский перевод: Галлагер Р., Теория информации и надежная связь, изд-во «Советское радио», 1974.
42. Kohlenburg A., Forney G. D., Jr., Convolutional Coding for Channels with Memory, *IEEE Trans.*, **IT-14**, 618—626 (1968).
43. Hsu H. T., Kasami T., Chien R. T., Error-Correcting Codes for a Compound Channel, *IEEE Trans.*, **IT-14**, 135—138 (1968).
44. Weldon E. J., Jr., New Generalization of the Reed-Muller Codes-Part II: Non-Primitive Codes, *IEEE Trans.*, **IT-14**, 199—206 (1968).
45. Robinson J. R., Частное сообщение, April 1973.

К главе 6

1. Wozencraft J. M., Sequential Decoding for Reliable Communication, MIT Research Lab. of Electronics, Tech. Report 325, Cambridge, Mass., 1957.
2. Shannon C. E., A Mathematical Theory of Communications, *BSTJ*, **27**, 379—423 (Part I), 623—656 (Part II), 1948; есть русский перевод: Шеннон К., Математическая теория связи, сб. *Работы по теории информации и кибернетике*, ИЛ, 1963, стр. 242—332.
3. Gallager R. G., A Simple Derivation of the Coding Theorem and Some Applications, *IEEE Trans.*, **IT-11**, 3—18 (1965); есть русский перевод: Галлагер Р., Простой вывод теоремы кодирования и некоторые применения, *Кибернетический сборник*, новая серия, вып. 3, изд-во «Мир», 1966, стр. 50—90.
4. Viterbi A. J., Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm, *IEEE Trans.*, **IT-13**, 260—269 (1967); есть русский перевод: Витерби А., Границы ошибок для сверточных кодов и асимптотически оптимальный алгоритм декодирования, сб. *Некоторые вопросы теории кодирования*, изд-во «Мир», 1970, стр. 142—165.
5. Shannon C. E., Gallager R. G., Berlekamp E. R., Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels, *Inform. and Control*, **10**, 65—103 (Part I), 525—552 (Part II), 1967; есть русский перевод: Шеннон К., Галлагер Р., Берлекэмп Е., Нижние границы вероятности ошибки для кодирования в дискретном канале без памяти, *Зарубежная радиоэлектроника*, **2** и **6**, стр. 52—81 и 41—64 (1968).
6. Wozencraft J. M., Reiffen B., Sequential Decoding, MIT Press, Cambridge, Mass., 1961; есть русский перевод: Возенкрафт Дж., Рейффен Б., Последовательное декодирование, ИЛ, 1963.
7. Jelinek F., Fast Sequential Decoding Algorithm Using a Stack, *IBM J. Res. Develop.*, **13**, 675—685 (1969).
8. Зигангиров К. Ш., Некоторые последовательные процедуры декодирования, *Проблемы передачи информации*, **2**, 13—24 (1966).
9. Savage J. E., Progress in Sequential Decoding, in: Advances in Communication Systems, Balakrisknan A. V., ed., Academic Press, N. Y., 1968.
10. Savage J. E., The Computation Problem with Sequential Decoding, MIT Research Lab. of Electronics, Tech. Report 439, 1965.
11. Gallager R. G., Information Theory and Reliable Communication, Wiley, N. Y., 1968; есть русский перевод: Галлагер Р., Теория информации и надежная связь, изд-во «Советское радио», 1974.
12. Jacobs I. M., Berlekamp E. R., A Lower Bound to the Distribution of Computation for Sequential Decoding, *IEEE Trans.*, **IT-13**, 167—174 (1967).
13. Yudkin H. L., Channel State Testing in Information Decoding, Ph. D. Dissertation, Dept. of Elec. Engrg., MIT, Cambridge, Mass., 1964.

14. Forney G. D., Jr., Convolutional Codes II: Maximum Likelihood Decoding and Convolutional Codes III: Sequential Decoding, Technical Report No. 7004-1, Information Systems. Lab., Stanford Univ., 1972.
15. Geist J. M., A Comparison of the Fano and Jelinek Sequential Decoding Algorithm, Technical Report EE 701, Univ. of Notre Dame, Notre Dame, Indiana, 1970.
16. Falconer D. D., A Hybrid Sequential and Algebraic Decoding Scheme, Ph. D. Thesis, Dept. of Elec. Engrg., MIT, Cambridge, Mass., 1966.
17. Massey J. L., Threshold Decoding, MIT Press, Cambridge, Mass., 1963; есть русский перевод: Месси Дж., Пороговое декодирование, изд-во «Мир», 1966.
18. Massey J. L., Some Algebraic and Distance Properties of Convolutional Codes, in: Error Correcting Codes, Mann H. B., ed., Wiley, N. Y., 1968.
19. Costello D. J., Jr., A Construction Technique for Random-Error-Correcting Convolutional Codes, *IEEE Trans.*, **IT-15**, 631—636 (1969).
20. Park J. H., Jr., Inductive Proof of an Important Inequality, *IEEE Trans.*, **IT-15**, 618 (1969).
21. Bussgang J. J., Some Properties of Binary Convolutional Code Generators, *IEEE Trans.*, **IT-11**, 90—99 (1965).
22. Lin S., Lyne H., Some Results of Binary Convolutional Code Generators, *IEEE Trans.*, **IT-13**, 134—139 (1967).
23. Fano R. M., A Heuristic Discussion of Probabilistic Decoding, *IEEE Trans.*, **IT-9**, 64—73 (1963); есть русский перевод: Фано Р., Эвристическое обсуждение вероятностного декодирования, сб. *Теория кодирования*, изд-во «Мир», 1964, стр. 166—198.
24. Wozencraft J. M., Jacobs I. M., Principles of Communication Engineering, Wiley, N. Y., 1968; есть русский перевод: Возенкрафт Дж., Джекобс И., Теоретические основы техники связи, изд-во «Мир», 1969.
25. Jelinek F., Probabilitic Information Theory, McGraw-Hill, N. Y., 1968.
26. Falconer D. D., Niessen C. W., Simulation of Sequential Decoding for a Telemetry Channel, Massachusetts Inst. of Technology, Research Lab. of Electronics, Quart. Progress Rep. 80, 183—193, 1966.
27. Viterbi A. J., Convolutional Codes and Their Performance in Communication Systems, *IEEE Trans.*, **CM-19**, 751—772 (1971).
28. Massey J. L., Costello D. J., Jr., Justesen J., Polynomial Weights and Code Construction, *IEEE Trans.*, **IT-19**, 101—110 (1973).

К главе 7

1. Киясу, Мурога, Теория информации, Серия «Современная прикладная математика», изд-во «Иванами», 1957.
2. Chien R. T., Coding for Error Control in a Computer-Manufacturer's Environment: a Foreword, *IBM J. Res. Develop.*, **14**, 342 (1970).
3. Grosch H. R. J., High Speed Arithmetic: the Digital Computer as a Research Tool, *J. Optical Soc. of America*, **43**, 306—310 (1953).
4. Кобаяси, Като, О повышении надежности полупроводниковой оперативной памяти с помощью кодов, исправляющих ошибки, Тезисы докладов национальной конференции общества связи, № 1215, 1971.
5. Holsinger J. L., Digital Communication over Fixed-Continuous Channels with Memory-with Special Application to Telephone Channels, MIT Lincoln Lab. Tech. Report 366, 1964.
6. Ивадари, Общие принципы проектирования систем связи с контролем ошибок, *Дэнси цусин гаккай ромбунси*, **54-A**, 87—95 (1971).
7. Cahn C. R., Performance of Digital Phase Modulation Communication Systems, *IRE Trans.*, **CS-7**, 3—6 (1959).

8. Иман, Катаока, Миякава, Многоступенчатое кодирование и его применение в системах связи с полифазной модуляцией, *Дэнси цусин гаккай ромбунси*, **54-A**, 597—604 (1971).
9. Wozencraft J. M., Kennedy R. S., Modulation and Demodulation for Probabilistic Coding, *IEEE Trans.*, **IT-12**, 291—297 (1966).
10. Jordan K. L., Jr., The Performance of Sequential Decoding in Conjunction with Efficient Modulation, *IEEE Trans. on Comm. Tech.*, **COM-14**, 283—297 (1966).
11. Levesque A. H., Recent Development in Error Control Techniques, Signal Processing Techniques in Digital Communication, 1969 Wescon Tech. Papers, 1969.
12. Bennet W. R., Darey J. R., Data Transmission, McGraw-Hill, N. Y., 1965.
13. Cacciamani E. R., Jr., The SPADE System as Applied to Data Communication and Small Earth Station Operation, COMSAT, *Tech. J.*, **1**, 171—182 (1971).
14. Сэкимото, Утино, Сабури, Система SPADE для спутниковой связи, Материалы исследований по системам связи научного общества электроники и связи, CS 71-131, 1972.
15. Кумагая, Предисловие к специальному выпуску, посвященному современной технике передачи информации, *Дэнси цусин гаккай ромбунси*, **53**, 11, 1466 (1970).
16. *IBM J. Res. Develop.*, **14** (1970).
17. *Tang D. T., Lum V. Y.*, Error Control for Terminals with Human Sperators, *IBM J. Res. Develop.*, **14**, 409—416 (1970).
18. Hsiao M. Y., Bossen B. C., Chien R. T., Orthogonal Latin Square Codes, *IBM J. Res. Develop.*, **14**, 390—394 (1970).
19. Iwadare Y., A Class of High-Speed Decodable Burst-Correcting Codes, *IEEE Trans.*, **IT-18**, 817—821 (Corresp.) (1972).
20. Chien R. T., Burst-Correcting Codes with High-Speed Decoding, *IEEE Trans.*, **IT-15**, 109—113 (1969).
21. Golomb S. W., ed., Digital Communications with Space Applications, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1964.

К главе 8

1. Von Neumann J., Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components, in: Automata Studies, Princeton Univ. Press, Princeton, N. J., 1956; есть русский перевод: Нейман Дж., Вероятностная логика и синтез надежных организмов из ненадежных компонент, сб. Автоматы, ИЛ, 1956, стр. 68—139.
2. Winograd S., Cowan J. D., Reliable Computation in the Presence of Noise, MIT Press, Cambridge, Mass., 1963; есть русский перевод: Виноград С., Коуэн Дж., Надежные вычисления при наличии шумов, изд-во «Наука», 1968.
3. Elias P., Computation in the Presence of Noise, *IBM J. Res. Develop.*, **2**, 346—353 (1958).
4. Diamond J. M., Checking Codes for Digital Computers, *Proc. of IRE*, **43**, 487—488 (1955).
5. Brown D. T., Error Detecting and Correcting Binary Codes for Atithmetic Operations, *IRE Trans.*, **EC-9**, 333—337 (1960).
6. Такаги, Лекции по теории чисел, изд-во «Томотати», 1931.
7. Виноградов И. М., Основы теории чисел, изд-во «Наука», 1965.
8. Reitwiesner G. H., Binary Arithmetic, in: Advances in Computers, Alt F. L., ed., Vol. 1, 232—308, Academic Press, N. Y., 1969.
9. Peterson W. W., Error-Correcting Codes, MIT Press and Wiley, N. Y., (1961).

10. Ерош И. Л., Ерош С. Л., Арифметические коды с исправлением многократных ошибок, *Проблемы передачи информации*, т. 3, вып. 4, 72—80 (1967).
11. Massey J. L., Survey of Residue Coding for Arithmetic Errors, International Computation Center Bulletin, UNESCO, Rome, Italy, **3**, 195—209 (1964).
12. Chiang A. C. L., Reed I. S., Arithmetic Norms and Bounds of the Arithmetic AN Codes, *IEEE Trans.*, **IT-16**, 470—476 (July 1970).

К главе 9

1. Mandelbaum D., Arithmetic Codes with Large Distance, *IEEE Trans.*, **IT-13**, 237—242 (1967).
2. Tsao-Wu N., Arithmetic Cyclic Codes, Dept. of Electrical Eng., Northeastern Univ., Boston, Mass., Part I of Communication Theory Group Report No. 10, Final Report AFCRL-68-0512, 1968.
3. Асано, Теория колец и идеалов, изд-во «Томотати», 1959.
4. Goto M., Fukumura T., The Distance of Arithmetic Codes, *Memoirs of the Faculty of Engineering*, Nagoya Univ., **20**, 474—482 (1968).
5. Chien R. T., Hong S. T., Preparata F. P., Some Results in the Theory of Arithmetic Codes, Coordinated Science Labs., Univ. of Illinois, Urbana, Report R-440, 1969.
6. Mandelbaum D., Arithmetic Error Detecting Codes for Communication Links Involving Computers, *IEEE Trans.*, **COM-13**, 165—17 (1965).
7. Атовара, Фукумура, Методы построения *AN*-кодов, исправляющих пачки ошибок, *Дэнси цусин гаккай ромбунси*, **50**, 2101—2107 (1967).
8. Massey J. L., Garcia O. N., Error-Correcting Codes in Computer Arithmetic, in: Advances in Information Systems Science, Tou J. T., ed., Vol. 4, Plenum Press, N. Y., 1971.
9. Avizienis A., A Study of the Effectiveness of Fault-Detecting Codes for Binary Arithmetic, Jet Propulsion Lab., Pasadena, Calif., Technical Report 32-711, 1965.
10. Chiang A. C. L., Reed I. S., Arithmetic Norms and Bounds of the Arithmetic AN Codes, *IEEE Trans.*, **IT-16**, 470—476 (July 1970).
11. Rao T. R. N., Garcia O. N., Cyclic and Multi-Residue Code for Arithmetic Operations, *IEEE Trans.*, **IT-17**, 85—91 (Jan. 1971).
12. Rao T. R. N., Error Coding for Arithmetic Processors, Academic Press, N. Y. and London, 1964.
13*. Кабатянский Г. А., О границах для числа кодовых слов в двоичных арифметических кодах, *Проблемы передачи информации*, **12**, вып. 4, 46—54 (1967).

Предметный указатель

Алгоритм Витерби 226
— Евклида 68
— Зигангирова — Джелинека 225
— Фано 224
AN-код 466, 479
— дополняющий 520
— исправляющий кратные ошибки 499
— — ошибки веса 1, 493
— — пачки ошибок 541
— методы декодирования 502
— обнаруживающий ошибки веса 1 493
— — пачки ошибок 541
— оптимальный 531
— простой 510
— сильно циклический 506
— совершенный
— — исправляющий ошибки веса 1 493
— циклический
— — разложение 509
Асимптотически хорошая последовательность кодов 204

Базис 80
— дополняющий 122

Вектор кодовый 128
— нулевой 78
— ошибок 134
Векторы линейно зависимые 79
— линейно независимые 79
— ортогональные 80
Вес 20
— арифметический 487
— минимальный 129
— модулярный 518
— Хэмминга 128
Вычет квадратичный 110

Геометрия аффинная 115
— евклидова 115
— проективная 118
Гиперплоскость 115
Гомоморфизм 57
Граница БЧХ 161
— Варшамова — Гилберта 140
— верхняя минимального расстояния *AN*-кода 527
— — Плоткина 32
— — Хэмминга 30
— — Элайса 34
— случайного кодирования 39
Группа 54
— абелева 55
— аддитивная 56
— бесконечная 58
— изоморфизм 58
— коммутативная 55
— конечная 58
— циклическая 65

Декодер 24
Декодирование дефинитное 270
— мажоритарное 179
— по максимуму правдоподобия 26
— пороговое 222
— последовательное 222
— с обратной связью 239
— списком 52
Делитель нуля 72
Длина двоичного *AN*-кода 480
— кода 23
Дистрибутивность 69

Закон взаимности 476

Идеал 70
— главный 507
Избыточность 13
— *AN*-кода 481
Изоморфизм 58
Индекс подгруппы 61

Информационный шумовой символ 243
Исправление ошибок 13
— — AN-кодом 490
Исправляемая конфигурация ошибок 532

Канал двоичный симметричный 26
— дискретный без памяти 38
Китайская теорема об остатках 542
Код арифметический 466
— блоковый 23
— Васильева 31
— Голея 31
— Гоппы 206
— групповой 128
— двойственный 129
— диффузный 223
— допускающий полную ортогонализацию 188
— Ивадари 223
— из последовательностей максимальной длины 34
— исправляющий ошибки 24
— каскадный 200
— квазисовершенный 136
— линейный 128
— — вычетный 466
— многоостаточный 542
— обнаруживающий ошибки 24
— ортогонализируемый 254
— перфорированный равномерный 293
— полиномиальный 196
— равновесный 16
— равномерный 223
— расширенный 21
— — евклидово-геометрический 197
— рекуррентный 222
— Рида — Маллера 189
— — обобщенный 189
— — — расширенный 189
— — циклический укороченный 189
— Рида — Соломона 166
— сверточный 222
— — самоортогональный 223
— — систематический 228
— систематический 23
— с максимально достижимым расстоянием 133
— совершенный 30
— укороченный 131, 161
— Файра 220
— Хэмминга 20
— — двоичный 158
— — обобщенный 134
— — циклический 164
— циклический 148
— — укороченный 161
— Шенгейма 31
— эквидистантный 34
— Юстесена 204
Кодер 23
Кодовое слово 23
— — AN-кода 479
— число AN-кода 479
Кодовый вектор 20
Кольцо 69
— вычетов 72
— коммутативное 69
— многочленов 70
— с единицей 69
— целых чисел по модулю 2^e — 1 506
Корень 84
Круговое число 477
k-пространство 115

Линейная комбинация 79
Локатор ошибок 170

Мажоритарная функция 179
Матрица Адамара 111
— порождающая 129
— проверочная 129
Метод декодирования, основанный на переборе 537
— — — — циклических сдвигах 534
— перемежения 218
Минимальное представление 485
Многоуровневая фазовая модуляция 432
Многочлен 82
— двойственный 103
— круговой 106
— Матсона — Соломона 184, 189
— минимальный 97
— неприводимый 84
— нормированный 84
— порождающий 150
— примитивный 98
— проверочный 151
— самодвойственный 103
Модуль 469

Наибольший общий делитель 468
Наименьшее общее кратное 467

Невычет квадратичный 110
Неортогональная проверка 184
Нормированный делитель 62
Нумератор 143

Область целостности 72
Обнаружение ошибок AN-кодом 490
Обобщенный метод чередования 271
Операция 54
Ортогональное дополнение 80
Ортогональность линейных e-сумм 180
Отображение биективное 57
— гомоморфное 57
— изоморфное 58
— инъективное 57
— сюръективное 57
Ошибка арифметическая 531
— веса t 26, 489
— исправляемая 532
— реально возможная 505

Пачка ошибок 28
— — типа B1 280
— — — B2 280
Подгруппа 59
— собственная 59
Подкод 128
— над подполем 195
— четный 131
Подполе 76
Подпространство 80
Показатель 470
Поле 72
— изоморфизм 74
— конечное 73
— простое 107
Полная ортогонализация 182
Полная система вычетов 469
Порождающее число AN-кода 479
Порядок 58, 64
Последовательность максимальной длины 159
Правило ортогонализации 238
Представление 66
— в виде степеней 95
— векторное 94
— минимального веса 482
— многочленное 94
Приведенная система вычетов 470
Примитивный корень степени d 476
Примитивный элемент 471
Проверочная сумма 180
Проверочный параллелограмм 285
Производная многочлена 87
Пространство векторное 78
Путь 301

Разложение мультипликативной группы 473
— на простые сомножители 468
Размерность 79
Разностное множество 121
Распределение весов 143
Распространение ошибок 239
Расстояние арифметическое 483
— конструктивное 164
— Ли 22
— минимальное 23
— — AN-кода 489
— — простого AN-кода 517
— модулярное 518
— свободное 471
— столбцевое 371
— Хэмминга 20
Расширение поля 76
— кода 131

Символ информационный 23
— Лежандра 475
— проверки на четность 18
— проверочный 23
Синдром 134, 229
— AN-кода 502
— усеченный 233
Система алгебраическая 54
Скалярное произведение 80
Скорость передачи 23
След 122
Смежный класс левый 59
— — правый 59
Составная проверка 239
Сравнимость по модулю 67
Стек-алгоритм 225
Структура циклического AN-кода 509
Структурная избыточность 466

Таблица групповая 57
— декодирования 24
Теорема Лагранжа 61
— Ферма 470
— Эйлера 470
Точка бесконечно удаленная 121

Укорачивание 161
Укорочение кода 131

Факторгруппа 62
Функция Кармайкла 472
— надежности 44
— производящая 91
— Эйлера 69, 468
— — мультипликативность 469

Характеристика поля 77

Центральный интервал 519
Цена пути 304
Цикл 103

Частное 467
Число информационных символов 23
— ортогональных проверок 241
— проверочных символов 20, 23
— простое 468
— составное 468

Элемент единичный 55, 73
— нулевой 73
— обратный 56
— примитивный 96

Оглавление

Предисловие редакторов русского издания 5
Предисловие авторов 7

1. Основные понятия теории кодирования 13

1.1. Коды, обнаруживающие и исправляющие ошибки 13
1.2. Блоковые коды. Систематические коды 21
1.3. Двоичный симметричный канал 26
1.4. Верхние границы для минимального расстояния кодов 28
1.4.1. Верхняя граница Хэмминга 30
1.4.2. Верхняя граница Плоткина 32
1.4.3. Верхняя граница Элайса 34
1.5. Теорема кодирования 38
1.5.1. Граница случайного кодирования 39
1.5.2. Свойства функции надежности 44
1.5.3. Граница сферической упаковки 50
1.5.4. Декодирование списком 52

2. Конечные поля 54

2.1. Группы 54
2.2. Кольца и поля 69
2.3. Векторные пространства 78
2.4. Многочлены 82
2.5. Конечные поля 90
2.6. Дополнительные сведения о конечных полях 110
2.6.1. Вычисления в конечных полях 110
2.6.2. Матрицы Адамара 111
2.6.3. Конечные геометрии 115
2.6.4. Разностные множества 121
2.6.5. Дополняющий базис 122
2.6.6. Некоторые понятия, необходимые для определения кодов Гоппы 125
Задачи 125

3. Линейные и циклические коды 128

3.1. Линейные коды 128
3.2. Методы декодирования линейных кодов 134
3.3. Нижняя граница Варшамова — Гилберта 140
3.4. Распределение весов 142
3.5. Циклические коды (I) 148
3.6. Циклические коды (II) 155
3.7. Укороченные коды 161

4. Важнейшие коды 163

4.1. Коды Боуза — Чоудхури — Хоквингема 163
4.2. Декодирование БЧХ-кодов 169
4.2.1. Основные этапы 169

4.2.2. Итеративный алгоритм Берлекэмпа 174
4.3. Методы мажоритарного декодирования 178
4.4. Многочлены Матсона — Соломона 184
4.5. Полиномиальные коды 188
4.5.1. Обобщенные коды Рида — Маллера 188
4.5.2. Полиномиальные коды и двойственные к ним коды 195
4.6. Каскадные коды и коды Юстесена 200
4.6.1. Каскадные коды 200
4.6.2. Коды Юстесена 204
4.7. Коды Гоппы 206
4.7.1. Определение 206
4.7.2. Метод декодирования 214
4.8. Коды, исправляющие пачки ошибок 218

5. Сверточные коды. I. Методы порогового декодирования 222

5.1. Общий обзор сверточных кодов 222
5.1.1. Метод порогового декодирования 222
5.1.2. Методы последовательного декодирования 223
5.1.3. Методы декодирования по максимуму правдоподобия . . . 225
5.2. Представление сверточных кодов 226
5.2.1. Представление с помощью операторов задержки 226
5.2.2. Представление с помощью матриц 230
5.3. Пример порогового декодирования 236
5.4. Принцип порогового декодирования 239
5.5. Самоортогональные коды 242
5.5.1. Определение самоортогонального кода 242
5.5.2. Простой пример 244
5.5.3. Коды, строящиеся с помощью простых совершенных разностных множеств 248
5.5.4. Примеры кодов 251
5.5.5 Оптимальность кодов 251
5.6. Ортогонализируемые коды 254
5.6.1. Простейшие примеры 254
5.6.2. Способ построения кодов 255
5.6.3. Примеры кодов 256
5.6.4. Связь с блоковыми кодами 256
5.6.5. Сопоставление с циклическими кодами, построенными на основе разностных множеств 259
5.7. Распространение ошибок 260
5.7.1. Анализ глубины распространения ошибок 261
5.7.2. Критерий устойчивости пороговой декодирующей логической схемы 264
5.7.3. Критерий, основанный на использовании функции Ляпунова 266
5.7.4. Дефинитное декодирование 270
5.8. Сверточные коды, исправляющие пачки ошибок 270
5.8.1. Методы чередования 271
5.8.2. Принцип построения кодов 273
5.8.3. Способ построения кодов 274
5.8.4. Конкретные коды 276
5.8.5. Оценки кодов и сравнение их параметров 277
5.8.6. Другие коды 279
5.9. Сверточные коды, исправляющие пачки ошибок и независимые ошибки (диффузные коды) 279
5.9.1. Два типа пачек ошибок 280
5.9.2. Корректирующая способность 280
5.9.3. Простой пример 281
5.9.4. Самоортогональные коды 282

5.10. Равномерные сверточные коды 284
5.10.1. Определение равномерного кода 284
5.10.2. Правило ортогонализации 285
5.10.3. Метод ортогонализации, обеспечивающий небольшую глубину распространения ошибок 288
5.10.4. Перфорированные равномерные коды 290
Задачи 293

6. Сверточные коды. II. Последовательное декодирование 298

6.1. Древовидные коды и принцип последовательного декодирования 299
6.1.1. Древовидные коды 299
6.1.2. Принцип последовательного декодирования 301
6.2. Алгоритм Фано 305
6.2.1. Цена пути 305
6.2.2. Поведение декодера 307
6.3. Среднее число операций при декодировании 311
6.3.1. Среднее число операций 311
6.3.2. Свойство независимости в древовидном коде 315
6.3.3. Верхняя граница для среднего числа операций 317
6.4. Распределение числа операций и вероятность переполнения буфера 321
6.4.1. Структура и функционирование буфера 322
6.4.2. Нижняя граница для $p[c \geqslant L]$ 323
6.4.3. Верхняя граница для $p[c \geqslant L]$ 328
6.4.4. Связь с вероятностью переполнения буфера 329
6.5. Вероятность необнаружения ошибки 334
6.6. Границы Витерби и декодирование по максимуму правдоподобия 338
6.6.1. Верхняя граница для вероятности ошибки 339
6.6.2. Нижняя граница для вероятности ошибки 344
6.7. Гибридные методы кодирования 348
6.7.1. Системы гибридного кодирования 348
6.7.2. Характеристики гибридного кодирования 353
6.8. Стек-алгоритм 358
6.9. Структура расстояний сверточных кодов 362
6.9.1. Верхняя граница Плоткина при декодировании с обратной связью 363
6.9.2. Нижняя граница Гилберта для сверточных кодов при декодировании с обратной связью 367
6.9.3. Верхняя и нижняя границы минимального расстояния при дефинитном декодировании 370
6.10. Коды, используемые при декодировании с обратной связью . . . 371
6.11. Коды, используемые при последовательном декодировании . . . 376
Задачи и упражнения 379

7. Реализация и применение кодов, исправляющих ошибки 383

7.1. Реализация кодов, исправляющих ошибки 383
7.1.1. Стандартные схемы 385
7.1.2. Кодеры циклических кодов 388
7.1.3. Декодеры циклических кодов 391
7.2. Реализация порогового декодирования 405
7.2.1. Кодеры 405
7.2.2. Декодер 410
7.2.3. Схема преобразования параллельных слов в последовательные и обратная схема 410
7.2.4. Реализация сверточных кодов, исправляющих пачки ошибок 411

7.3. Обсуждение связи теории кодирования с реальными техническими проблемами 412
7.3.1. Задача теории кодирования 415
7.3.2. Стоимость и надежность 417
7.3.3. Применение в вычислительных системах 418
7.4. Различные предположения, используемые в теории кодирования 420
7.4.1. Дискретные величины 421
7.4.2. Методы принятия решений 421
7.4.3. Расстояние Хэмминга 424
7.4.4. Положительные стороны теории кодирования 425
7.5. Применения в системах связи метода повторной передачи . . . 426
7.5.1. Системы с полной обратной связью и системы с дублированием передачи 426
7.5.2. Обнаружение ошибок 427
7.6. Применения в системах связи кодов, исправляющих ошибки . . 428
7.6.1. Модуляция и кодирование 428
7.6.2. Вероятность ошибки при использовании алгебраических кодов 430
7.6.3. Многоуровневая фазовая модуляция и кодирование . . . 432
7.6.4. Применения кодов в космических и спутниковых системах связи 437
7.6.5. Основные понятия о проектировании систем связи с помехоустойчивым кодированием 448
7.6.6. Проблемы, возникающие при проектировании систем связи с помехоустойчивым кодированием информации 450
7.6.7. Пример применения порогового декодирования в спутниковой связи 451
7.7. Применение в системах обработки информации 453
7.7.1. Применение кодов при общении человека с машиной . . . 454
7.7.2. Коды на основе ортогональных латинских квадратов . . . 457
7.7.3. Коды, исправляющие пачки ошибок и допускающие быстрое декодирование 461
Задачи 465

8. Коды для арифметических устройств 466

8.1. Основные понятия теории чисел 467
8.1.1. Делимость целых чисел 467
8.1.2. Наименьшее общее кратное и наибольший общий делитель 467
8.1.3. Простые числа и разложение на простые сомножители . . 468
8.1.4. Функция Эйлера 468
8.1.5. Сравнимость и классы вычетов 469
8.1.6. Теоремы Эйлера и Ферма 470
8.1.7. Показатели экспоненты и примитивные корни 470
8.1.8. Мультипликативные группы и их разложение 473
8.1.9. Квадратичные вычеты и символ Лежандра 475
8.1.10. Круговые многочлены и их сомножители 476
8.1.11. Минимальное представление дроби 478
8.2. Определение AN-кода 479
8.3. Арифметический вес и арифметическое расстояние 481
Алгоритм нахождения представления, удовлетворяющего условию M 481
8.4. Минимальное расстояние и корректирующая способность AN-кода 489
8.5. Обнаружение и исправление независимых ошибок веса 1 . . . 493
8.6. AN-коды, исправляющие кратные ошибки 499
8.7. Синдромы и методы декодирования AN-кодов 502

9. Циклические *AN*-коды 506

9.1. Структура циклических *AN*-кодов 506
9.1.1. Определение циклического *AN*-кода 506
9.1.2. Кольцо целых чисел по модулю $2^e - 1$ и его идеалы . . . 506
9.1.3. Задание циклических кодов 508
9.1.4. Разложение и структура циклических *AN*-кодов 509
9.2. Минимальное расстояние циклических *AN*-кодов 517
9.2.1. Минимальное расстояние простых кодов 517
9.2.2. Минимальное расстояние *AN*-кодов, удовлетворяющих специальным условиям 521
9.2.3. Минимальное расстояние циклических *AN*-кодов (*B* — простое число) 524
9.2.4. Минимальное расстояние циклических *AN*-кодов (*B* — составное число) 528
9.3. Декодирование циклических *AN*-кодов 531
9.3.1. Вычеты по модулю *A* и конфигурации ошибок 531
9.3.2. Метод декодирования, основанный на циклических сдвигах вычетов 534
9.3.3. Метод декодирования, основанный на переборе 537
9.4. Дополнение 540

Приложения 545

Приложение 1. Разложение чисел вида $2^n - 1$ на простые множители и таблица неприводимых многочленов 545
Приложение 2. Параметры двоичных БЧХ-кодов в узком смысле длины 1023 и менее 546

Литература 556

Предметный указатель 568

ИБ № 579

Т. Касами, Н. Токура, Ё. Ивадари, Я. Инагаки

ТЕОРИЯ КОДИРОВАНИЯ

Редактор Л. Якименко Художник А. Быков Художественный редактор В. Бисенгалиев
Технический редактор Г. Алюлина Корректор В. Соколов

Сдано в набор 07.09.77. Подписано к печати 16.02.78. Формат 60×90 1/16. Бум. тип. № 1. Литературная гарнитура. Высокая печать. 18,00 бум. л., 36,00 печ. л. Уч.-изд. л. 34,25. Зак. № 761. Цена 2 р. 70 к.

Издательство «Мир». Москва, 1-й Рижский пер., 2

Ордена Трудового Красного Знамени Ленинградская типография № 2 имени Евгении Соколовой Союзполиграфпрома при Государственном комитете Совета Министров СССР по делам издательств, полиграфии и книжной торговли. 198052, Ленинград, Л-52, Измайловский проспект, 29.

Т. КАСАМИ, Н. ТОКУРА,
Ё. ИВАДАРИ, Я. ИНАГАКИ

ТЕОРИЯ КОДИРОВАНИЯ

2.170.